

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

**ANALYSIS OF THE AERODYNAMIC ORBITAL
TRANSFER CAPABILITIES OF A WINGED RE-ENTRY
VEHICLE**

by

John P. Pienkowski

September 2002

Thesis Advisor:
Second Reader:

Stephen A. Whitmore
Michael G. Spencer

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 2002	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE: Analysis of the Aerodynamic Orbital Transfer Capabilities of a Winged Re-Entry Vehicle			5. FUNDING NUMBERS	
6. AUTHOR(S) John P. Pienkowski				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) <p>The ability to perform an inclination change maximizes the maneuverability of an orbiting space vehicle. Most maneuvers utilize a combined plane change and orbital transfer to the new orbit. This costs more in terms of energy and fuel than an in-plane change of orbits. The amount of ΔV and fuel required for such an energy-intensive inclination change exceeds the benefit of performing the maneuver. However, this paper demonstrates that a winged re-entry vehicle, based on the currently proposed X-37, has the necessary thrust to change planes and then perform an in-plane transfer to achieve a new orbit. Using SIMULINK™ and LABVIEW™ simulation tools, this research found that the use of the aerodynamic lift of a winged re-entry vehicle produced more than 12° of inclination change with the minimal ΔV achievable. Through small orbital maneuvers and atmospheric re-entry, the aerodynamics of the lift vector demonstrated that the spacecraft retained sufficient energy to prevent perigee collapse using an orbital regulation code to control throttle setting.</p>				
14. SUBJECT TERMS Winged Re-entry Vehicle, Orbital Mechanics, Orbital Transfer, Perigee Collapse, Orbital Regulator, Aerodynamic Forces, Fuel Optimal, Inclination Change			15. NUMBER OF PAGES 123	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

**ANALYSIS OF THE AERODYNAMIC ORBITAL TRANSFER CAPABILITIES
OF A WINGED RE-ENTRY VEHICLE**

John P. Pienkowski
Lieutenant, United States Navy
B.S., United States Naval Academy, 1995

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SPACE SYSTEMS OPERATIONS

from the

**NAVAL POSTGRADUATE SCHOOL
September 2002**

Author: John P. Pienkowski

Approved by: Stephen A. Whitmore
Thesis Advisor

Michael G. Spencer
Second Reader

Rudolf Panholzer
Chairman, Space Systems Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The ability to perform an inclination change maximizes the maneuverability of an orbiting space vehicle. Most maneuvers utilize a combined plane change and orbital transfer to the new orbit. This costs more in terms of energy and fuel than an in-plane change of orbits. The amount of ΔV and fuel required for such an energy-intensive inclination change exceeds the benefit of performing the maneuver. However, this paper demonstrates that a winged re-entry vehicle, based on the currently proposed X-37, has the necessary thrust to change planes and then perform an in-plane transfer to achieve a new orbit. Using SIMULINK™ and LABVIEW™ simulation tools, this research found that the use of the aerodynamic lift of a winged re-entry vehicle produced more than 12° of inclination change with the minimal ΔV achievable. Through small orbital maneuvers and atmospheric re-entry, the aerodynamics of the lift vector demonstrated that the spacecraft retained sufficient energy to prevent perigee collapse using an orbital regulation code to control throttle setting.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	HISTORICAL PERSPECTIVE	1
A.	INTRODUCTION	1
B.	BRIEF HISTORY OF WINGLESS LIFTING BODIES.....	2
1.	The Early Years (1960 – 1975).....	2
2.	The Later Years (1975 – 1998).....	6
C.	CURRENT PROPOSED WINGED RE-ENTRY VEHICLES.....	9
1.	X-38.....	9
2.	X-40A.....	10
3.	X-37.....	11
II.	MISSION DESCRIPTION	13
A.	BACKGROUND	13
1.	Reusable Launch Vehicle Study.....	13
2.	X-37 Program Definitions	14
B.	PLANE CHANGE MANEUVERS.....	16
1.	Simple, Combined, and Hohmann Transfers	16
2.	X-37 Based Vehicle Maneuver Description.....	19
III.	SIMULATION EQUATIONS.....	21
A.	COMPUTATIONAL COORDINATE SYSTEM.....	21
B.	EQUATIONS OF MOTION.....	22
C.	EXTERNAL FORCES ACTING ON THE SPACECRAFT.....	23
D.	TRANSFORMATION FROM GAUSSIAN TO INERTIAL COORDINATE SYSTEMS.....	27
E.	MODELING THE EARTH’S ATMOSPHERE.....	31
F.	CALCULATING THE WIND-RELATIVE TRAJECTORY	34
IV.	ORBITAL REGULATOR DESIGN.....	37
A.	ENERGY ANALYSIS.....	37
1.	Perigee Collapse	37
2.	Regulator State Equation.....	39
B.	CONSTRAINED CONTROL EQUATIONS.....	42
1.	Hamiltonian Functional.....	42
2.	Interpretation of Constrained Optimal Control.....	46
3.	Summary	46
V.	SIMULATION TOOLS.....	49
A.	MATLAB™/SIMULINK™.....	49
B.	LABVIEW™	52
VI.	DISCUSSION OF RESULTS	55
A.	ANALYSIS OF THE AERODYNAMIC LIFT VECTOR.....	55
1.	Tuning the Orbital Regulator.....	55
2.	Attitude Determination.....	58
B.	DV REQUIRED FOR MAXIMUM INCLINATION CHANGE	63
VII.	CONCLUSION.....	69
	LIST OF REFERENCES	73
	APPENDICES.....	75
	INITIAL DISTRIBUTION LIST.....	105

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1.	M2-F1, M2-F2, and HL-10. (After: NASA Dryden Photo Gallery)	3
Figure 2.	X-24A and X-24B. (After: NASA Dryden Photo Gallery)	4
Figure 3.	X-24A, M2-F3, and HL-10. (From: NASA Dryden Photo Gallery)	5
Figure 4.	Orbiter Separation. (From: NASA Dryden Photo Gallery)	7
Figure 5.	Bell Nozzle and Linear Aerospike Engine. (After: NASA Dryden Photo Gallery)	8
Figure 6.	X-33 and X-34. (After: NASA Dryden Photo Gallery).....	9
Figure 7.	X-38. (After: NASA Dryden Photo Gallery).....	10
Figure 8.	X-40A. (After: NASA Dryden Photo Gallery)	11
Figure 9.	X-37. (After: NASA Dryden Photo Gallery).....	12
Figure 10.	Simple Plane Change. (From: Whitmore Lecture Notes).....	17
Figure 11.	Combined Plane Change. (From: Whitmore Lecture Notes)	17
Figure 12.	X-37 Maneuver Description. (From: Whitmore Lecture Notes)	20
Figure 13.	Gaussian Coordinate System. (From: Whitmore Lecture Notes).....	21
Figure 14.	Vector Force Diagram. (From: Whitmore Lecture Notes)	26
Figure 15.	Inertial Coordinate System. (After: Whitmore Lecture Notes).....	28
Figure 16.	Temperature vs. Altitude.	33
Figure 17.	Pressure vs. Altitude.	33
Figure 18.	Effect of Drag on Apogee. (After: Whitmore Lecture Notes).....	37
Figure 19.	Perigee Collapse. (After: Whitmore Lecture Notes)	38
Figure 20.	Apogee vs. Perigee Curve. (After: Whitmore Lecture Notes).....	38
Figure 21.	Differential Energy. (After: Whitmore Lecture Notes)	41
Figure 22.	SIMULINK™ Model.	51
Figure 23.	LABVIEW™ Center Panel.	53
Figure 24.	Normalized Max L/D vs. Angle of Attack.	56
Figure 25.	Q Ratio vs. Inclination Change.....	57
Figure 26.	SIMULINK™ Run 1.	59
Figure 27.	LABVIEW™ Run 1.	59
Figure 28.	Comparison between Run 2 Simulations.....	60
Figure 29.	Comparison between Run 3 Simulations.....	61
Figure 30.	Comparison of Runs 2, 4, and 5.	62
Figure 31.	Fuel Load vs. Inclination Change.	64
Figure 32.	Inclination Change vs. Time.	65
Figure 33.	Altitude vs. Time.	65
Figure 34.	Delta-V vs. Time.	68

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1.	X-37 Main Engine Statistics. (After: Andrews Space & Technology)	14
Table 2.	Out-of-Plane Change Options. (After: Understanding Space)	19
Table 3.	Classical Orbital Elements.....	27
Table 4.	Atmospheric Constants. (After: U.S. Standard Atmosphere, 1976)	31
Table 5.	Gain Ratio Results.	57
Table 6.	Effects of Initial Orbit on Maximum Plane Change.	62
Table 7.	Example Simulation Initial Conditions.	63

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF SYMBOLS

r	Radius
V	Velocity
DV	Change in Velocity
m	Mass
H	Geometric altitude
T	Temperature
P	Pressure
ρ	Density of air
c	Sonic velocity
a	Semi-major axis
i	Inclination
v	True anomaly
w	Argument of perigee
W	Right ascension of the ascending node
ε	Specific mechanical energy
m	Gravitational parameter of Earth
g_c	Gravitational acceleration at sea level
g	Ratio of the Specific Heat of Air at constant pressure to the Specific Heat of Air at constant volume
M	Mach number
\bar{q}	Dynamic pressure
C_L	Coefficient of lift
C_D	Coefficient of drag
I_{sp}	Specific Impulse
g_{pa}	Flight path angle
θ	Pitch
f	Roll
p	Co-state variable (adaptive gain)
q_1, q_2	Gain variables

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Stephen “Tony” Whitmore, for his enthusiasm and perseverance in this project. His guidance from the idea’s conception to its conclusion proved to be extraordinary and is greatly appreciated. In addition, I would like to thank my co-advisor, Dr. Michael Spencer, for his patience while working with me through the initial programming stages of the analysis. The success of this thesis is largely due to their support and I am thankful for the opportunity to learn from the both of them.

THIS PAGE INTENTIONALLY LEFT BLANK

I. HISTORICAL PERSPECTIVE

A. INTRODUCTION

The science of wingless flight seeks to demonstrate that light-weight lifting bodies supply a tremendous amount of data in the area of low lift-to-drag ratio, unpowered, horizontal-landing spacecraft. This section briefly outlines the chronological history of wingless lifting bodies and their contribution to the development of an unpowered, horizontal-landing spacecraft. The culmination of years of research manifests itself in today's current projects, the most promising being winged re-entry vehicles and their applicability for a future Military Spaceplane (MSP). Studies performed by the U.S. Air Force and NASA have demonstrated the desire to expand orbital operations flexibility. One approach taken includes the utilization of the X-37 Orbital Maneuvering Vehicle (OMV) to lower its orbital altitude and perform plane change maneuvers using aerodynamic lift; therefore, saving fuel and maximizing its ability to change orbits for the cost of a single launch. The concept of using a highly throttleable and restartable rocket motor, as a part of a feedback loop that stabilizes the orbit during aero-maneuvering, has never been published in open literature. The objective of this research project begins with the assessment of an X-37-based vehicle that uses aerodynamic forces to change its orbital inclination. It will develop a strategy to optimize the performance of these maneuvers for maximum inclination change and minimum fuel consumption. Using computer simulation software to design tools to test the orbital regulation strategy, the project concluded that maximum inclination change occurred at the spacecraft's maximum lift-to-drag ratio and high angles of roll ($>70.0^\circ$). Within 10 orbits, the X37 based vehicle achieved 12 to 16 degrees of inclination change for the minimum amount of fuel expended.

B. BRIEF HISTORY OF WINGLESS LIFTING BODIES

1. The Early Years (1960 – 1975)

In August of 1963, the M2-F1 lightweight lifting body program saw its first successful air-tow flight test at the Dryden Flight Research Center (Reed, pg. 49). These first flight tests provided much needed technical and political confidence to the early pioneers of lifting-body research. The M2-F1 program had profound effects on every follow-on space vehicle, including the Space Shuttle Program (Reed, pg. 63). Two more programs followed the M2-F1 design during the period of 1966-1968: the HL-10 and M2-F2. The HL-10's shape resembled a hydroplane racing boat because the Langley engineers considered horizontal landings on the water (Reed, pg. 71). The HL-10 was unique in that it had a high lift-to-drag ratio as compared to the M2-F1. In addition, the HL-10 used a "dive bomber" landing approach. It made a very steep approach and then flared and lowered its landing gear in the last few moments before touchdown. While there was risk from a last minute mechanical failure, this procedure allowed for greater accuracy during an unpowered approach (Reed, pg. 125). The M2-F2 vehicle nearly resembled the M2-F1 except that it lacked the outer horizontal elevons. These were removed due to heating concerns and shockwave impingement during re-entry from space. More importantly, the M2-F2 had a forward canopy for increased visibility during the landing procedure. While its lift-to-drag ratio was between the M2-F1's and the HL-10's, it would eventually weigh 10 times as much as the M2-F1 (Reed, pg. 80). **Figure 1** illustrates the schematics of the M2-F1, M2-F2, and HL-10 vehicles.

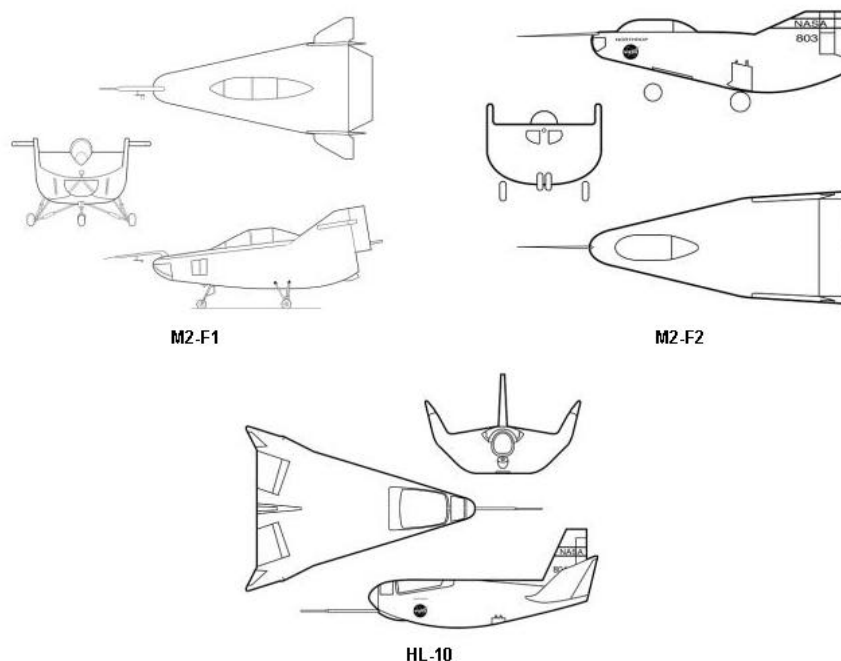


Figure 1. M2-F1, M2-F2, and HL-10. (After: NASA Dryden Photo Gallery)

In 1969, a new dawning in powered flight tests occurred. Specifically, the rollout and initial unpowered flight tests of the X-24A were witnessed. Powered test flights occurred between mid-March 1970 to June 1971. Nearly 200 pounds lighter than the HL-10, the X-24A sought to perform maneuvers near the proposed maximum Mach speeds in order to gather data for precision powered control. The X-24A set lifting-body speed and altitude records at Mach 1.6 and 71,000ft. The ability to select one of four individual XLR-11 rocket engines allowed for controlled thrust levels (Reed, pg. 139). With 28 successful flights, the X-24A validated the concept that a space vehicle could be landed unpowered. Since the X-24A had proven itself as a reliable lifting body, its shape was selected later as the basis for the International Space Station's lifeboat program, the Crew Return Vehicle (CRV). Discussed later in this section, the X-38 project became the prototype design for the CRV. **Figure 2** illustrates the 3-view schematic of the X-24A and X-24B vehicles.

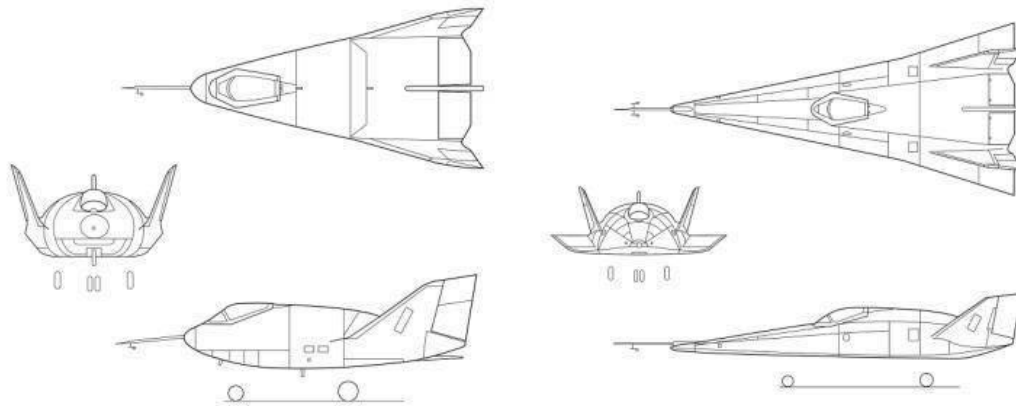


Figure 2. X-24A and X-24B. (After: NASA Dryden Photo Gallery)

The next program, M2-F3, saw its debut in June 1970 (Reed, pg.148). The essential difference between the M2-F2 and M2-F3 was the installation of a center fin. This added better roll control (Reed, pg. 114). 700 pounds heavier than the X-24A, the M2-F3 design added many significant design features. Reaction control jets powered by hydrogen peroxide eliminated the need for elevons and rudders. Only one flap on the vehicle's bottom for longitudinal trim would be required. The pilot would rely on one control system from orbit to landing (Reed, pg. 116). Therefore, the M2-F3 was considered the "purest" form of lifting body design. It had no horizontal projections or tail surfaces, which could be construed as small wings (Reed, pg. 144). The M2-F3 is featured in the center of **Figure 3** along with the X-24A and HL-10 on the desert lakebed at the Dryden Flight Research Center.



Figure 3. X-24A, M2-F3, and HL-10. (From: NASA Dryden Photo Gallery)

By August 1973, the X-24B began its first unpowered glide flight tests. Wrapping a new shell around the X-24A vehicle comprised its new shape. Essentially, engineers converted the original bulbous, teardrop shape of the X-24A into a “flatiron” shape with a rounded top, flat bottom, double-delta winged platform that ended with a pointed nose. This added about 1800 pounds of weight and two outboard ailerons for roll control (Reed, pg. 171). Throughout 1974 and 1975, the X-24B set many new altitude and speed records for lifting bodies, one at 74,100 feet and another at Mach 1.75 (Reed, pg. 173). Of all the lifting bodies tested over a twelve-year period, test pilots considered the X-24B to have the best control characteristics during landing without power. By comparison, the X-24A had the highest landing lift-to-drag ratio of 4.5. The next highest ratio was the X-24A at 4.0, then the HL-10 at 3.6, and the lowest was the M2-F3 at 3.1 (Reed, pg. 175).

Since the Space Shuttle Program was well into design phase by 1975, the X-24B was used to simulate unpowered landing tests at Edwards’ concrete runway. These precision touchdowns illustrated to senior shuttle program managers that vehicles configured for relatively low lift-to-drag ratios could achieve accurate, unpowered landings. This convinced shuttle authorities that the air-breathing jet engines, originally planned for the orbiters, could be eliminated for a significant benefit in weight savings and payload capacity (Reed, pg. 175). All of the aforementioned vehicles contributed

greatly to the technology and development of future designs for space transportation. Their legacy has had a significant impact on how flight tests for the Space Shuttle and follow-on spacecraft will be performed. During the 1980's and 1990's, lifting body technology developments spread internationally as Russia, Japan, and the European Space Agency (ESA) began to design and test their own vehicles. By adapting its legacy craft with updated technology enhancements, the United States also stepped forward with its new designs for follow-on transportation vehicles in order to expand orbital operations in space.

2. The Later Years (1975 – 1998)

Through the founding research conducted on lifting bodies, the Space Shuttle Program conducted its Approach and Landing Test Program (ALT) in 1977. The goal of the nine month program was to demonstrate that the orbiter, designated OV-101 *Enterprise*, could fly in the atmosphere and land like an airplane without the aid of powered flight. The ALT program was comprised of several phases, both ground and flight tests. Ground tests validated that the Shuttle Carrier Aircraft (SCA), a modified 747, could taxi and brake with the *Enterprise* mated to the top. Five captive carry flights with the unmanned *Enterprise* mounted to the SCA assessed the structural integrity and handling capabilities of the mated aircraft. Three manned captive flights followed and the astronaut crew aboard the orbiter tested its flight control systems in preparation for free flight tests.

The next phase of testing involved five free flights where the astronaut crew separated the *Enterprise* from the SCA and performed maneuvers to a landing at Edwards Air Force Base. These five free flights each contained milestones. The first four landings occurred on a dry lakebed, while the last took place on the concrete runway at Edwards AFB simulating the conditions as a return from space. **Figure 4** shows the separation of the orbiter and the SCA. In addition, the last two free flights were made with the aerodynamic tail cone removed to simulate an actual return from space. The free flight tests demonstrated subsonic flight mechanics and the orbiter's ability to approach

and land safely with various weight and center-of-gravity configurations. The ALT program proved a number of technologies and processes. Crew and engineers learned the value of low-speed aero maneuvers and the procedures necessary to safely and successfully conduct atmospheric test flights of a space shuttle orbiter.



Figure 4. Orbiter Separation. (From: NASA Dryden Photo Gallery)

In 1994, the X-33 design by Lockheed Martin's Skunk Works resulted from a NASA proposal to develop a single-stage-to-orbit vehicle to replace the Space Shuttle. Incorporating lifting-body technology, the wingless craft was designed for vertical takeoff and horizontal landing. Its critical accomplishment resided in its engine design. Utilizing the linear aerospike engine, the X-33's trailing edge contained seven aerospike engines. Essentially a conventional rocket engine turned inside out, the linear aerospike engine maximizes efficiency throughout its flight path. For example, each of the seven engines can be individually throttled to maintain maximum efficiency at various altitudes. Conversely, a conventional rocket nozzle is designed for its highest level of efficiency at a single altitude. **Figure 5** demonstrates the differences between a conventional rocket nozzle and the linear aerospike engine.

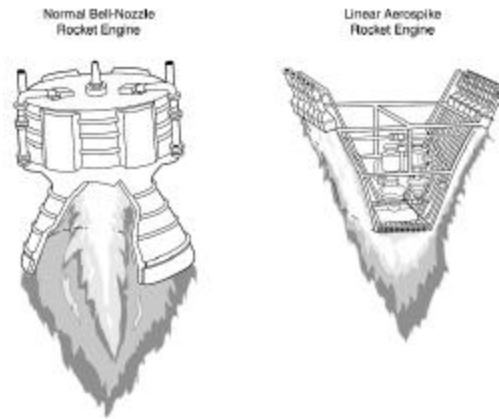


Figure 5. Bell Nozzle and Linear Aerospike Engine . (After: NASA Dryden Photo Gallery)

The X-33 prototype for the future of Reusable Launch Vehicles (RLVs) offered dramatically changed technologies and materials as compared to the Space Shuttle (Reed, pg. 184):

- Single-Stage-To-Orbit; no Solid Rocket Boosters (SRBs) or External Fuel Tank (EFT)
- Metal heat shield; eliminating thousands of hours of maintenance on ceramic tiles
- Liquid Oxygen (LOX)/Liquid Hydrogen (LH₂) for all propellants; no hypergolics
- Self-contained canister for payload bay; stand alone testing
- Efficient linear aerospike engine; no gimbaled rocket nozzle

As an interim step to test materials and concepts for the X33 prototype, NASA and the Orbital Sciences Corporation joined to develop the X34 Project. Powered by a rocket engine using kerosene and LOX, the X-34 was designed to be dropped from a Lockheed L-1011 airliner that Orbital Sciences had configured for its Pegasus winged booster launches (Larson, pg. 1). Also designed for horizontal landing, the X-34 researched advanced avionics to gain valuable early flight data for use in the X-33

program (Gonzales, et al, pg. 50). **Figure 6** compares the X-33 and X-34 designs. Unfortunately, both the X-33 and X-34 projects lost their funding because of budgetary constraints. However, the technologies developed, especially that of the linear aerospike engine, have been archived and passed forward to future designs and tests.



Figure 6. X-33 and X-34. (After: NASA Dryden Photo Gallery)

C. CURRENT PROPOSED WINGED RE-ENTRY VEHICLES

1. X-38

Full-scale, unpiloted free-flight drop tests of the X-38 began in March 1998. Using the X-24A's design concept, the goal of the X-38 Project is to develop the technology for a prototype Crew Return Vehicle (CRV) for the International Space Station (ISS). However, due to scale-backs on the International Space Station, the X-38 project had its funding placed in standby in 2001. Successful scale-model test flights have demonstrated the capabilities of the new technologies used in the flight profile:

- Expendable deorbit engine jettisoned as a module
- Unpowered approach like the Space Shuttle
- Steerable parafoil parachute to control its final landing descent
- Final landing in the desert on skids rather than wheeled landing gear

- More durable thermal protection system – special ablative coating applied to the thermal tiles
- New software operating systems, electromechanical actuators, and video equipment flight tested aboard the Space Shuttle and other NASA experiments (Pike, pg. 1).

Figure 7 illustrates the 3-view schematic of the X-38 vehicle. Note the similarities between the shape of the X-38 and the X-24A designs.

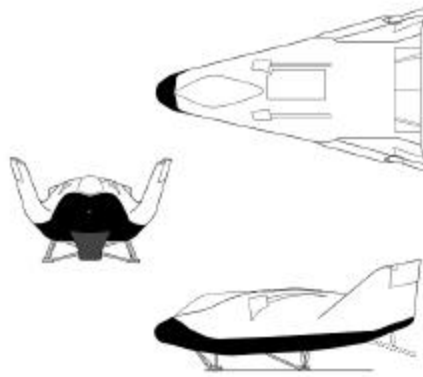


Figure 7. X-38. (After: NASA Dryden Photo Gallery)

2. X-40A

The X-40A project manifested itself from the U.S. Air Force's desire to develop a military space plane. The Air Force Research Laboratory laid the groundwork for this Integrated Technology Testbed (ITT) project. Their design entails a small powered space vehicle technology demonstrator known as the Space Maneuver Vehicle (SMV). The project goal sees the SMV as a two-stage-to-orbit vehicle as well as a reusable satellite with a variety of payloads. Its small size and ability to shift orbital inclination and altitude allows it to reposition for tactical advantage or geographic sensor placement. Envisioned to dwell on orbit for up to one year, the interchangeability of the SMV's payloads permits a wide variety of missions. For example, tactical reconnaissance,

deployment of satellite constellations, and a time-critical communications relay platform are some of the missions the SMV could perform.

The SMV is under development by Boeing Phantom Works and an 85% scale unpowered testbed, the X-40A, was rolled out in September 1997. It is 22 feet long, has a wingspan of 12 feet, and weighs about 2600 pounds. The first flight of the X-40A achieved its goal on 11 AUG 1998 when it was released by an UH-60 Black Hawk helicopter. From 9,000 ft, it made a controlled landing using its own avionics and on-board systems. The final concept consists of a reusable “mini-spaceplane” that is carried to hypersonic speeds by a suborbital reusable first stage (Pike, pg. 1). **Figure 8** shows the X-40A vehicle during a test flight.



Figure 8. X-40A. (After: NASA Dryden Photo Gallery)

3. X-37

The X-40A SMV program shifted its focus when the Air Force teamed with NASA and Boeing Company, Inc. to begin the X-37 project. The USAF loaned the X-40A test vehicle to NASA to use as a test article for the similarly designed X-37. To move the X-37 Project forward while the test vehicle was being built, Phase 1 of the X-37 Project consisted of several free flight tests using the X-40A vehicle. Phase 2 will conduct X-37 unpowered flights and Phase 3 will be orbital test flights. The X-37 vehicle will be 27.5 feet long and have a wingspan of 15 ft. The payload bay is designed

to be 7 feet in length by 4 feet in width. The X-37 will be powered by two AR2-3A rocket engines fueled with hydrogen peroxide and JP-8 that can produce 7000 pounds of thrust.

Between early 2001 and May 2001, Boeing conducted seven successful test flights using the X-40A vehicle. Each flight progressively demonstrated the vehicle's ability to control its descent autonomously from various altitudes and maneuvering profiles (Cast, et al, pg. 1). During Phase 2, the operational X-37 will be carried under a B-52 aircraft to a suborbital altitude where it will be dropped for atmospheric test flights. The X-37 project is currently funded for two flights. Once developed, the X-37 will remain on orbit for up to two days on the first mission and up to 21 days for the second mission. When on orbit, the X-37 will test space vehicle technologies, including a solar array system developed by the U.S. Air Force. The X-37 will be designed for 20 flights and 420 days of cumulative on-orbit duration ("2000 Reusable Launch Vehicle Programs and Concepts." pg. 21). **Figure 9** shows an artist's conception of the X-37 operational vehicle in flight.

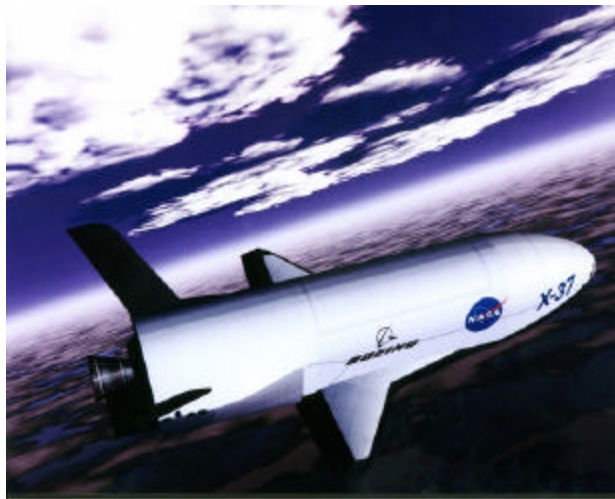


Figure 9. X-37. (After: NASA Dryden Photo Gallery)

II. MISSION DESCRIPTION

A. BACKGROUND

1. Reusable Launch Vehicle Study

In August 2001, the U.S. Air Force Space Command performed an assessment of military spaceplanes and reusable launch vehicles (RLVs). The goals of the study were to identify the operational utility, applicability, science, and technology maturity of the X-33 and X-37 programs as well as identify other possible options. The conclusions of the working panel recommended that a closer partnership develop between NASA and the Air Force concerning NASA's Space Launch Initiative (SLI) and RLV technologies. The most apparent alternative is a Two-Stage-to-Orbit (TSTO) option with a mix of expendable and reusable vehicles in order to cover the range of missions and operations. For example, operational considerations of the working group included overland launches, refurbishment, rapid payload integration, rapid on-orbit checkout, and orbital operational flexibility. In addition, payloads carried aboard such missions included imaging sensors (SIGINT, MASINT, radar, etc.) or microsatellites.

The working panel concluded that the X-33 and X-37 programs provide limited advances in the enabling technologies the Air Force desired. Both programs proved valuable as technology demonstrators and contributed toward the understanding of cost, schedule, performance, and integration issues. However, of the two programs, the working panel recommended not to pursue the X-33 prototype. The bottom line result was the X-33 seemed least likely to offer an achievable concept that could lead to an operational vehicle. While the X-37 demonstrated marginal utility as a military spaceplane, the positive advances in the arena of thermal protection system, autonomous guidance, re-entry profile, recovery and reconstitution procedures revealed the need for further analysis (Anarde Study).

2. X-37 Program Definitions

From the Anarde Council conclusions and recommendations, gaps still existed in the capabilities of future RLV programs. Operational concepts and requirements definitions were considered immature. Most importantly, the need continues to expand orbital operations flexibility. The objective of the X-37 program manifests itself as a reusable orbital maneuvering vehicle to reduce the cost of space transportation via flight demonstration and mature advanced technology. The X-40A test program validated the ability to perform autonomous landings through various flight profiles (Cast, et al, pg. 1). The key component of the X-37 vehicle focuses on its propulsion system. The main engine of the X-37 is a Rocketdyne AR2-3A originally designed as a rocket thruster to give the F-111 the ability to launch from an aircraft carrier. With variable throttle and the capability for rapid multiple restarts, its versatility gives the X-37 a valuable asset for the propulsion system. **Table 1** lists the details of the AR2-3A main engine.

Propellants:	90% H ₂ O ₂ & JP-8
Oxidizer:	2454 lbm H ₂ O ₂
Fuel:	327 lbm JP-8
Propellant Mass Fraction:	0.3900
Nominal Thrust:	3300 lbf
Vacuum Isp:	240.8 sec
Total Available DV:	2544.86 ft/sec
Throttle Range:	10% to 110%

Table 1. X-37 Main Engine Statistics. (After: Andrews Space & Technology)

The following sections of this report will demonstrate that the ability of a spacecraft to perform either an in-plane or an out-of-plane orbital change requires a significant amount of fuel. Accordingly, this fuel must be carried aboard the spacecraft and increase the gross weight of the launcher and vehicle at liftoff. This equates to larger launch costs and heavier boosters for a single launch. The measure of the fuel required,

engine efficiency, and amount of plane change a space vehicle can perform is categorized in the term Total Available ΔV , or simply ΔV . For example, the ΔV of the Space Shuttle only allows it to perform an out-of-plane inclination change of one degree per launch. The Space Shuttle would not have much fuel remaining to perform any in-plane maneuvers. Therefore, the Space Shuttle must launch directly into the orbit at which it will operate.

For an X-37 based vehicle as noted in **Table 1**, the total available ΔV is excessive for an in-plane maneuver; however, it is not enough for a significant plane change. The question is, what if an alternate method could be used instead of a pure fuel-intensive maneuver? An X-37 based vehicle has the ability to use its shape as a lifting body to perform an aerodynamic maneuver to alter its flight path and thus change inclination. *If* an X-37 based vehicle can be demonstrated to have the potential to change up to 10° orbital inclination, then it can achieve a dual use status: a single spacecraft performing two missions for the price of one launch. This cost-savings approach matches the need to expand orbital operations flexibility. Therefore, this thesis research assesses the feasibility of using aerodynamic forces to change the orbital inclination of an X-37 based vehicle. Furthermore, developing optimized strategies to perform these maneuvers remains a goal for this project. The approach taken begins with the development of a real-time, “piloted” simulation that allows a rapid evaluation of a wide variety of candidate maneuvers and trajectories. The simulation serves to provide insight as to which parameters are important to the problem. Batch simulations programmed in MATLABTM capable of performing extended Monte-Carlo analysis are cross validated with the real-time simulation. The “piloted” simulation tool can be used for generating feasible starting trajectories for follow-on optimization codes, such as DIDO and POST. The formulation discussed in this report reflects the codes used in the batch MATLABTM programs, but is nearly identical to those used in the real-time simulation. The next section describes the maneuvers necessary to conduct an out-of-plane inclination change.

B. PLANE CHANGE MANEUVERS

1. Simple, Combined, and Hohmann Transfers

The Hohmann Transfer, which uses an elliptical transfer orbit, remains the most fuel-efficient method for changing orbits within the same plane. Simply put, the Hohmann Transfer changes the in-plane size of the orbit. However, if a spacecraft needs to change its inclination, it would have to perform either a simple or combined plane change. The out-of-plane maneuver takes much more energy and thus fuel than the in-plane change. A simple plane change only alters the tilt, or inclination, of the orbit, as illustrated in **Figure 10**. The Earth is spinning around the z-axis and the initial orbit is along the y-axis at some initial velocity. Using plane geometry to solve the isosceles triangle, the relationship for ΔV is solved in Equation (1).

$$\Delta V_{simple} = 2V_1 \sin\left(\frac{\Delta i}{2}\right) \quad (1)$$

where:

ΔV_{simple}	= Velocity change for a simple plane change (km/s)
$V_1 = V_2$	= Velocities in the initial and final orbits (km/s)
Δi	= Plane change angle (deg or rad)

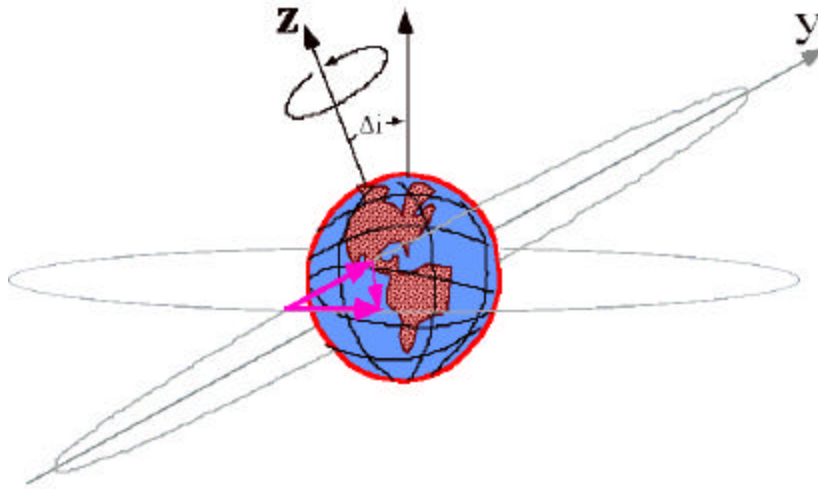


Figure 10. Simple Plane Change. (From: Whitmore Lecture Notes)

To change the inclination and size of the orbit, then a combined plane change is the easiest method. **Figure 11** shows the vector diagram of a combined plane change.

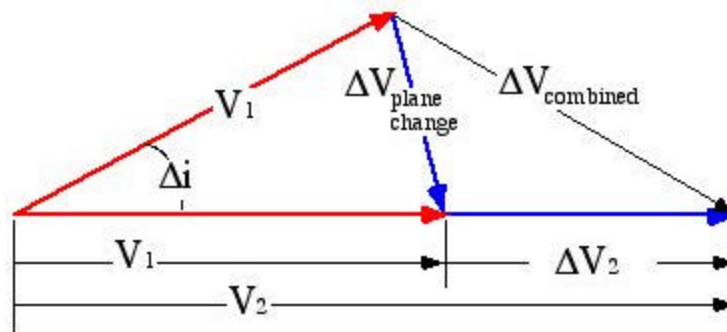


Figure 11. Combined Plane Change. (From: Whitmore Lecture Notes)

$DV_{combined}$ is the vector sum of the simple plane change, DV_{simple} , and the increase of the orbit's size, DV_2 . The first burn, DV_1 , places the spacecraft into its elliptical transfer orbit and DV_2 changes the size and circularizes the final orbit. These three ΔV burns make up a triangle with $DV_{combined}$ as the third side. Equation (2) uses the law of cosines to solve for $DV_{combined}$.

$$\Delta V_{combined} = \sqrt{(V_1)^2 + (V_2)^2 - 2V_1V_2 \cos(\Delta i)} \quad (2)$$

where:

$$\begin{aligned} \mathbf{DV}_{combined} &= \text{Velocity change for a combined plane change (km/s)} \\ V_1 &= \text{Velocity in the initial orbit (km/s)} \\ V_2 &= \text{Velocity in the final orbit (km/s)} \\ \mathbf{Di} &= \text{Plane change angle (deg or rad)} \end{aligned}$$

Applying these equations to an example for changing orbital inclination from a low-Earth orbit at 28.5° (latitude of the Kennedy Space Center) to a higher orbit at 0° inclination (equatorial), one can prove that the combined plane change maneuver is more fuel efficient than the simple plane change with multiple Hohmann Transfers. Equations (3) to (7) conceptualize the solutions for ΔV_{total} using the following values:

$$R_{orbit\ 1} = 6619 \text{ km}$$

$$R_{orbit\ 2} = 6798 \text{ km}$$

$$i_{orbit\ 1} = 28.5^\circ$$

$$i_{orbit\ 2} = 0^\circ$$

$$a_{transfer} = \frac{R_{orbit1} + R_{orbit2}}{2} \quad (3)$$

$$\mathbf{e}_N = -\frac{\mathbf{m}}{2a_N} \quad (4)$$

$$V_{orbitN} = \sqrt{2 \left(\frac{\mathbf{m}}{R_{orbitN}} + \mathbf{e}_{orbitN} \right)} \quad (5)$$

$$\Delta V_N = \left| V_{transfer} - V_{orbitN} \right| \quad (6)$$

$$\Delta V_{Hohmann} = \Delta V_1 + \Delta V_2 \quad (7)$$

where:

$a_{transfer}$ = Semimajor axis of the transfer orbit (km)

R_{orbit} = Magnitude of the spacecraft's position vector (km)

e_N = Specific mechanical energy of the spacecraft (km^2/s^2) at orbit N

μ = Gravitational parameter of the Earth, $3.986 \times 10^5 \text{ km}^3/\text{s}^2$

ΔV_N = Velocity change from the transfer orbit N into orbit N (km/s)

$\Delta V_{Hohmann}$ = Total ΔV needed for Hohmann Transfer (km/s)

Table 2 summarizes four cases explaining the relationship between simple and combined plane change maneuvers. Case 4 proves to be the most fuel-efficient because it starts with a Hohmann Transfer ΔV_1 burn in the lower orbit and then completes a combined plane change maneuver at the apogee (lower energy) of the higher orbit (Sellers, pg 205).

CASE1	CASE 2	CASE 3	CASE 4
Simple plane change of 28.5° , then Hohmann transfer, ΔV_1 and ΔV_2	Hohmann Transfer, ΔV_1 and ΔV_2 , then simple plane change of 28.5°	Combined plane change at perigee of transfer orbit, then ΔV_2 of Hohmann Transfer	ΔV_1 of Hohmann Transfer, then combined plane change at apogee of transfer orbit
$\Delta V_{\text{simple}} = 3.82 \text{ km/s}$ (orbit 1)	$\Delta V_{\text{Hohmann}} = 0.19 \text{ km/s}$	$\Delta V_{\text{combined}} = 3.84 \text{ km/s}$	$\Delta V_1 = 0.1 \text{ km/s}$
$\Delta V_{\text{Hohmann}} = 0.19 \text{ km/s}$	$\Delta V_{\text{simple}} = 3.72 \text{ km/s}$	$\Delta V_2 = 0.09 \text{ km/s}$	$\Delta V_{\text{combined}} = 3.77 \text{ km/s}$
$\Delta V_{\text{total}} = 4.01 \text{ km/s}$	$\Delta V_{\text{total}} = 3.91 \text{ km/s}$	$\Delta V_{\text{total}} = 3.93 \text{ km/s}$	$\Delta V_{\text{total}} = 3.87 \text{ km/s}$

Table 2. Out-of-Plane Change Options. (After: Understanding Space)

2. X-37 Based Vehicle Maneuver Description

This research project endeavors to prove an alternative to the expenditure of precious fuel using Hohmann Transfers and combined plane changes to alter the inclination of a spacecraft. An X-37 based vehicle will utilize the aerodynamic forces of

the lift and drag vectors of its winged lifting body and short engine thrusts to change its inclination over a period of a few orbits. At perigee, drag forces acting on the spacecraft will cause it to descend into the upper atmosphere. The spacecraft uses aerodynamic lift to alter its flight path with enough energy to maintain its orbital velocity. With a small boost from its main engine, the vehicle regains its original perigee altitude at a slightly new inclination. After many orbits repeating this process, the spacecraft will have changed its inclination several degrees. **Figure 12** illustrates the simplified description of the maneuver. When the drag forces cause the spacecraft to descend at perigee, the instantaneous altitude of both the perigee and apogee lowers. To prevent the perigee altitude from collapsing and forming an unrecoverable energy state, an orbital regulator is programmed into the maneuver. Using constrained control equations, the orbital regulator directs the main engine to fire short impulsive burns to re-boost the spacecraft, adding energy and thus increasing the perigee and apogee altitudes to nominal values. The following sections describe the programming codes and equations of motion necessary to perform this maneuver.

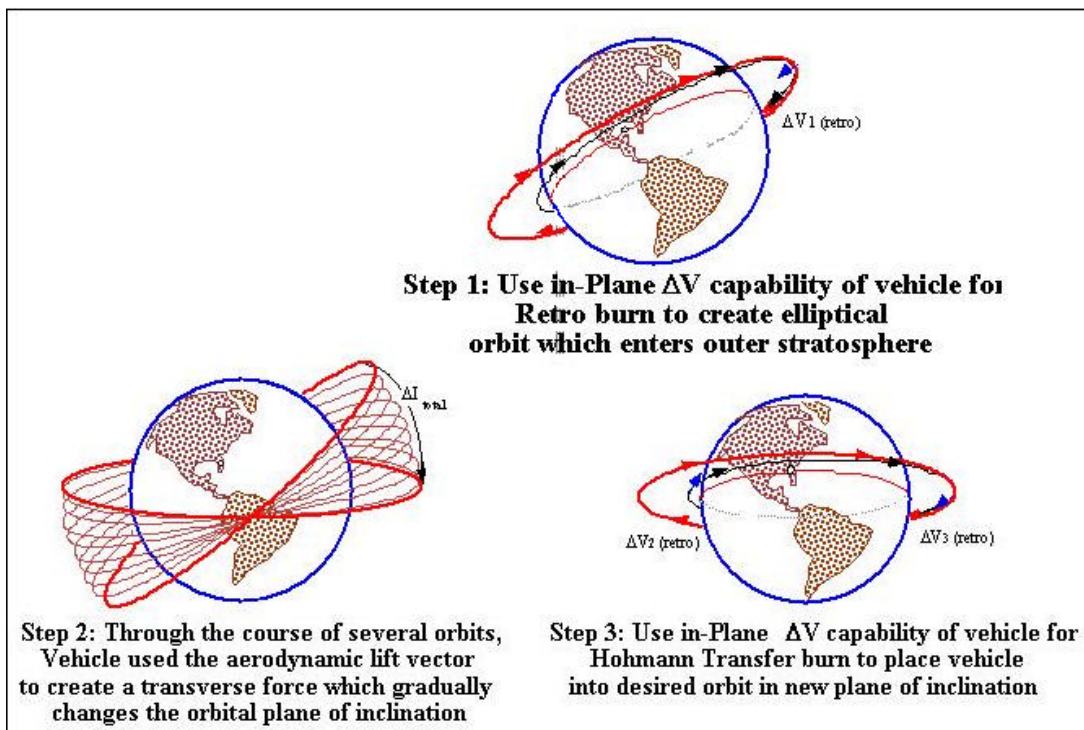


Figure 12. X-37 Maneuver Description. (From: Whitmore Lecture Notes)

III. SIMULATION EQUATIONS

A. COMPUTATIONAL COORDINATE SYSTEM

The differential equations used to describe the orbital motion of the spacecraft are derived using the *satellite* or *Gaussian* coordinate system (Vallado, pg. 162). In this coordinate system, the *r*-component points away from the center of the Earth in a radial direction, the *v*-component is perpendicular to the radial direction and points in the direction of travel of the spacecraft, and the *i*-component completes the right-handed orthogonal coordinate system. This coordinate systems stays fixed to the spacecraft at all times, and the *i*-coordinate is always perpendicular to the instantaneous orbital plane. The Gaussian coordinate system is depicted in **Figure 13**.

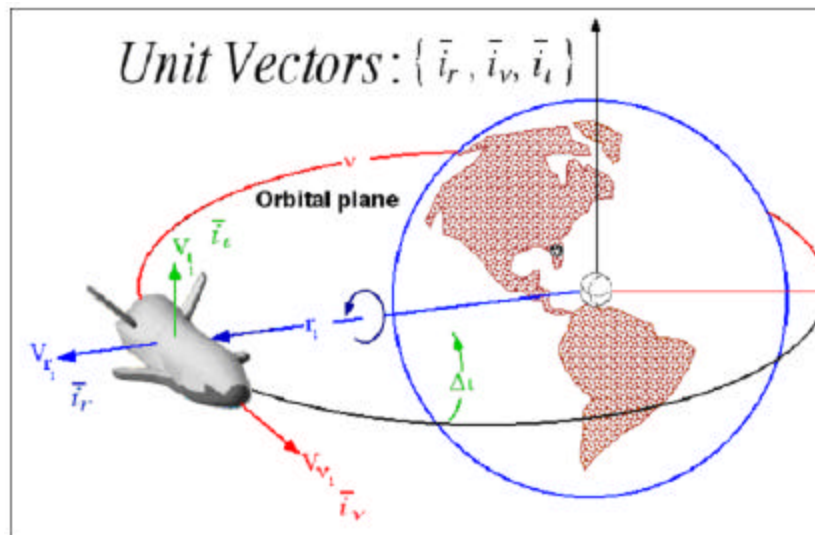


Figure 13. Gaussian Coordinate System. (From: Whitmore Lecture Notes)

B. EQUATIONS OF MOTION

Since the coordinate system is moving with the orbiting spacecraft, Equations (8) and (9) account for the effects of the angular motion of the spacecraft (Freidberger, pg. 797).

$$\bar{V} = \frac{\partial r}{\partial t} + \mathbf{v} \times \bar{r} = \dot{r}_r + \begin{bmatrix} \dot{r} \\ \dot{\mathbf{n}}_r \\ 0 \end{bmatrix} \quad (8)$$

$$\bar{a} = \frac{\partial \bar{V}}{\partial t} + \mathbf{v} \times \bar{V} = \begin{bmatrix} \dot{V}_r \\ \dot{V}_n \\ 0 \end{bmatrix} + \begin{bmatrix} -\dot{\mathbf{n}}V_n \\ \dot{\mathbf{n}}V_r \\ iV_n \end{bmatrix} \quad (9)$$

where:

- \dot{r} = Change in radial position with respect to time
- $\dot{\mathbf{n}}_r$ = Change in the direction of travel with respect to time
- \dot{V}_r = Change in velocity in radial direction with respect to time
- \dot{V}_n = Change in velocity in the direction of travel with respect to time

Combining Equations (8) and (9) achieves the dynamical system in Equation (10).

$$\bar{a} = \frac{1}{m} \begin{bmatrix} F_r \\ F_n \\ F_i \end{bmatrix} = \begin{bmatrix} \dot{V}_r \\ \dot{V}_n \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{V_n^2}{r} \\ \frac{V_n V_r}{r} \\ iV_n \end{bmatrix} \quad (10)$$

The left hand side of Equation (10) can be rewritten with the state variables to be propagated as the equations of motion in Equation (11).

$$\begin{bmatrix} \dot{r} \\ \dot{\mathbf{n}} \\ \dot{i} \\ \dot{V}_r \\ \dot{V}_n \end{bmatrix} = \begin{bmatrix} V_r \\ \frac{V_n}{r} \\ -\frac{1}{m} \frac{L \sin f}{V_n} \\ +\frac{V_n^2}{r} \\ -\frac{V_n V_r}{r} \end{bmatrix} \quad (11)$$

where:

- L = Lift force felt on the spacecraft (N)
- ϕ = Roll angle (deg)
- m = Spacecraft mass (kg)

C. EXTERNAL FORCES ACTING ON THE SPACECRAFT

The kinematics routine, called *forces*, compiles the data received from the *atmosv7*, *aero1*, and *orb3* subroutines to form the overarching program for the final product. First, it calls upon the *liftdrag2* subroutine to gather the necessary coefficients of lift and drag, C_L and C_D . Subsonic and supersonic data tables stored in the background of the MATLAB™ macro program supply the values needed for *liftdrag2*, which employs a two-dimensional interpolation routine to choose the best coefficient for use in Equations (12) and (13). The aerodynamic performance data used in this analysis was derived from analytical extrapolations of data from wind tunnel tests conducted by the Boeing Company (Seal Beach, CA). Due to proprietary data rights retained by Boeing, the specific aero data will not be presented in this thesis report. Appendix H contains the code for the *forces* subroutine and Appendix I lists the *liftdrag2* code. In order to compute the absolute lift and drag force acting on the vehicle, the lift and drag coefficients must be de-normalized by multiplying through by a reference area and the local dynamic pressure, \bar{q} . Equations (12) and (13) compute these parameters in the *forces* routine.

$$L = \bar{q} \cdot A_{ref} \cdot C_L \quad (12)$$

where:

- L = External lift force on a winged vehicle (N)
- \bar{q} = Dynamic pressure (pascals)
- A_{ref} = Reference area of an X-37-based vehicle (m²)
- C_L = Coefficient of Lift

$$D = \bar{q} \cdot A_{ref} \cdot C_D \quad (13)$$

where:

- D = External drag force on a winged vehicle (N)
- \bar{q} = Dynamic pressure (pascals)
- A_{ref} = Reference area of an X-37-based vehicle (m²)
- C_D = Coefficient of Drag

The last external force exerted on the spacecraft stems from the main engine. From **Table 1**, the AR2-3A main engine has a nominal thrust of 3300lbf and when multiplied by the throttle setting and then converted to metric results in Equation (14).

$$F_{thrust} = throt \cdot F_{nom} / 100 \quad (14)$$

where:

- F_{thrust} = Thrust from the main engine (N)
- $throt$ = Throttle setting programmed by the orbital regulator (%)
- F_{nom} = Nominal thrust of AR2-3A engine (N)

Since the engine will be allowed to fire during this simulation, the mass of the vehicle will no longer remain constant. The change of mass is accounted in a basic manipulation of the rocket equation listed in Equation (15). Thus, the mass flow rate of the fuel can be determined. Noting the value of the specific impulse, I_{sp} , from **Table 1**, Equation (16)

solves mass flow rate by dividing the thrust by the product of the I_{sp} and the local gravitational acceleration, g' , at the specific altitude.

$$I_{sp} = \frac{F_{thrust}}{\dot{m} \cdot g'} \quad (15)$$

$$\dot{m} = \frac{F_{thrust}}{I_{sp} \cdot g'} \quad (16)$$

Based on the variables formulated in Equation (11), the equations of motion derive from the external forces exerted on the spacecraft based on its roll, pitch, and angle of attack as demonstrated in Equation (17).

$$\begin{bmatrix} F_r \\ F_n \\ F_i \end{bmatrix} = \begin{bmatrix} L \cos \mathbf{g}_{fpa} \cos \mathbf{f} - (F_{grav} + D \sin \mathbf{g}_{fpa}) + F_{thrust} \sin \mathbf{q} \\ -(D \cos \mathbf{g}_{fpa} + L \sin \mathbf{g}_{fpa} \cos \mathbf{f}) + F_{thrust} \cos \mathbf{q} \\ -L \sin \mathbf{f} \end{bmatrix} \quad (17)$$

where:

- F_r = External force in the radial direction (N)
- F_n = External force in the direction of travel in the orbit (N)
- F_i = External force into the inclination plane (N)
- L = External lift force on a winged vehicle (N)
- D = External drag force on a winged vehicle (N)
- F_{grav} = Gravitational force at the specific altitude (m/s^2)
- F_{thrust} = Thrust from the main engine (%)
- ϕ = Roll angle (deg)
- θ = Pitch angle (deg)
- γ_{fpa} = Flight path angle (deg)

Figure 14 illustrates the relationship between the external forces. The angular difference between the direction of travel, denoted by the V_p vector, and the orientation of the nose of the spacecraft determines the flight path angle, γ_{fpa} .

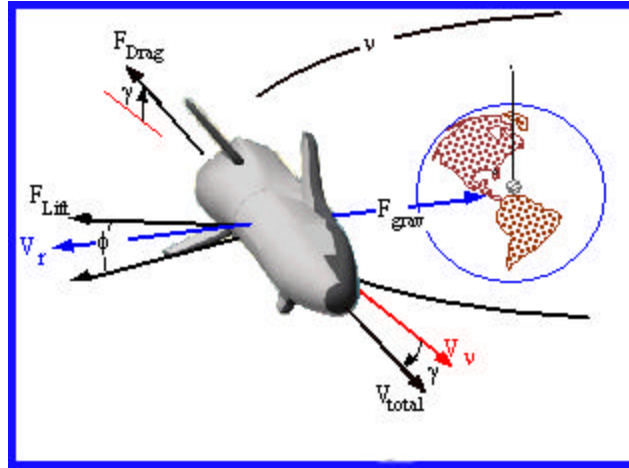


Figure 14. Vector Force Diagram. (From: Whitmore Lecture Notes)

Armed with the data collected through *forces* and the general equations for the external forces exerted on the spacecraft, the final equations of motion fall into place. The terms describe the accelerations felt in the three inertial directions, the change in position, the change in velocity, and the mass flow rate. The first part to the radial acceleration term, \dot{V}_r , accounts for the centrifugal acceleration felt by a body moving around the Earth. The second part uses the external force in the radial direction, F_r . The next term, \dot{V}_n , accounts for both the coriolis acceleration and the external acceleration felt in the direction of travel in the orbit. The other terms are listed in Equation (18). The external forces are divided by the spacecraft mass to keep them in terms of acceleration. J_2 perturbations are not included in these equations because their effects are negligible over the time the spacecraft spends in orbit. A more detailed analysis following this research would include J_2 perturbations to tune the orbital regulator and optimal control constraints.

$$\begin{bmatrix} \dot{V}_r \\ \dot{V}_n \\ \dot{i} \\ \dot{r} \\ \dot{\mathbf{n}} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} \left(\frac{V_n^2}{r} - \frac{\mathbf{m}}{|r|^2} \right) + \left(\frac{L \cos \mathbf{g}_{fpa} \cos \mathbf{f} - D \sin \mathbf{g}_{fpa} + \frac{F_{thrust} \sin \mathbf{q}}{m}}{m} \right) \\ - \left(\frac{V_r \cdot V_n}{r} \right) + \left(\frac{F_{thrust} \cos \mathbf{q}}{m} - \frac{D \cos \mathbf{g}_{fpa} + L \sin \mathbf{g}_{fpa} \cos \mathbf{f}}{m} \right) \\ - \frac{L \sin \mathbf{f}}{m \cdot V_n} \\ V_r \\ V_n \\ - \frac{F_{thrust}}{g' \cdot I_{sp}} \end{bmatrix} \quad (18)$$

D. TRANSFORMATION FROM GAUSSIAN TO INERTIAL COORDINATE SYSTEMS

Because significant non-conservative external forces, i.e. lift, drag, and thrust will be acting on the spacecraft, the instantaneous orbit will no longer remain constant. Thus, the instantaneous orbit must be computed from the Gaussian or in-plane velocity and position vectors. The set of six orbital elements used to describe the instantaneous orbit with respect to the inertial coordinate system are delineated in **Table 3**.

<u>Parameter</u>	<u>Symbol</u>	<u>Definition</u>
Semi-major axis	a	Defines size of orbit
Eccentricity	e	Defines shape of orbit
Inclination	i	Defines orientation of orbit with respect to Earth's equatorial plane
Argument of Perigee	ω	Defines perigee of orbit
Right Ascension of the Ascending Node	Ω	Defines location of the ascending/descending orbit nodes with respect to the Vernal Equinox (First Point of Aries)
True Anomaly	v	Defines location of spacecraft within the orbit with respect to the perigee

Table 3. Classical Orbital Elements.

These orbital parameters and the inertial coordinate system, are depicted in **Figure 15**. The x-axis points in the direction of the Vernal Equinox. The z-axis passes through the Earth's north pole and the y-axis completes the right-handed coordinate system. N1 and N2 mark the positions of the Ascending and Descending nodes. Thus, ω measures from the x-axis (Vernal Equinox) to the y-axis (RAAN). The Argument of Perigee measures from the nodes to the perigee point (periapsis). Inclination measures the tilt of the orbit away from the equatorial plane.

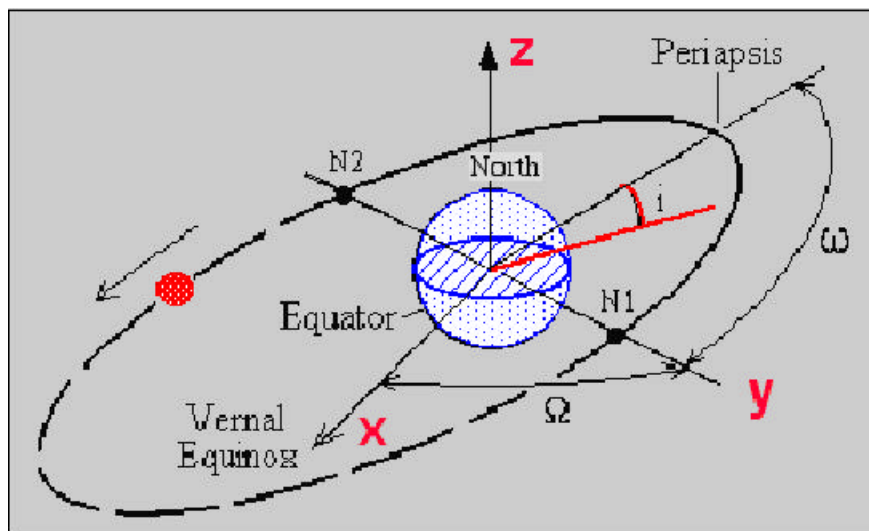


Figure 15. Inertial Coordinate System. (After: Whitmore Lecture Notes)

The first step in the orbit-determination process is to transform the in-plane position and velocity vectors to the inertial reference frame. The first piece, *orb1*, takes the initial inputs of true anomaly, inclination, position, and velocity and performs matrix algebra to transform the outputs into state vector format of current position and velocity. The second orbital subroutine, *orb2*, performs the matrix operations that transform the current position and velocity state vectors back to the inertial plane by calling on the subroutine, *rotate*. The last subroutine, *orb3*, takes the values from the inertial plane and calculates all of the orbital elements. Equations (19) and (20) show the generalized

matrices for the position and velocity vectors. All four subroutines are listed in Appendices D through G.

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{bmatrix} r \cos \mathbf{n} \\ r \sin \mathbf{n} \cdot \cos \mathbf{n} \\ r \sin \mathbf{n} \cdot \sin \mathbf{n} \end{bmatrix} \quad (19)$$

$$\text{where } R_{vect} = \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix}$$

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} V_r \cos \mathbf{n} - V_n \sin \mathbf{n} \\ (V_r \sin \mathbf{n} + V_n \cos \mathbf{n}) \cos i - V_i \sin i \\ (V_r \sin \mathbf{n} + V_n \cos \mathbf{n}) \sin i + V_i \cos i \end{bmatrix} \quad (20)$$

$$\text{where } V_{vect} = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$$

Orb2 performs the rotation matrix operations in order to convert the x-y-z coordinates into inertial coordinates for ease of calculations. Equation (21) rotates the Argument of Perigee, \mathbf{w} , in the orbital plane about the z-axis toward the Line of Nodes in the clockwise ($-\mathbf{w}$) direction.

$$\mathbf{w}_M = \begin{bmatrix} \cos \mathbf{w} & -\sin \mathbf{w} & 0 \\ \sin \mathbf{w} & \cos \mathbf{w} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (21)$$

Equation (22) rotates the Inclination, i , orbital plane about the x-axis toward the equatorial plane in the clockwise ($-i$) direction.

$$I_M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & -\sin i \\ 0 & \sin i & \cos i \end{bmatrix} \quad (22)$$

Equation (23) rotates Right Ascension of the Ascending Node, \mathbf{W} , about the z axis in the direction of the vernal equinox in the clockwise ($-\mathbf{W}$) direction.

$$\Omega_M = \begin{bmatrix} \cos \Omega & -\sin \Omega & 0 \\ \sin \Omega & \cos \Omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (23)$$

The final rotation is completed through matrix algebra shown in Equation (24) and the conversion to the inertial plane is represented by Equation (25).

$$M_R = [\Omega_M][I_M][\mathbf{w}_M] \quad (24)$$

$$\begin{aligned} R_{inert} &= M_R \cdot R_{vect} \\ V_{inert} &= M_R \cdot V_{vect} \end{aligned} \quad (25)$$

The remaining portion of *orb3* calculates the orbital elements semi-major axis, eccentricity, inclination, true anomaly, right ascension of the ascending node, argument of perigee, period, specific angular momentum, and the line of nodes. As follow-on computations progressed, perigee and apogee altitudes as well as the final ΔV result were added to the end of *orb3* to complete the analysis.

E. MODELING THE EARTH'S ATMOSPHERE

The effects of lift and drag on the vehicle are greatly dependent on the position within the Earth's atmosphere. Therefore, a static atmospheric model based on the 1976 Standard Atmosphere was developed, called *atmosv7*. The basis for this program allows for the evaluation of atmospheric parameters for a winged reentry vehicle. Therefore, given an altitude referenced from the center of the Earth, the program evaluates various atmospheric parameters and produces a graphic display based on an iterative model.

Table 4 delineates the values used in the subroutine code that is listed in Appendix A.

SYMBOL	VALUE	DESCRIPTION
R^*	$8.31432 \times 10^3 \frac{N \cdot m}{kmol \cdot K}$	Gas Constant
M_o	$28.9644 \frac{kg}{kmol}$	Molecular Weight of Air – Standard Temperature and Pressure
R_{air}	R^*/M_o	Gas Constant of Air
g'	$9.80665 \frac{m}{s}$	Gravitational Acceleration at sea level
γ	1.4	Ratio of Specific Heat of Air at constant pressure to the Specific Heat of Air at constant volume
ρ	$\frac{P_{mb}}{R_{air} \cdot T_{amb}}$	Density of Air
c	$\sqrt{I \cdot R_{air} \cdot T_{amb}}$	Sonic Velocity of Air

Table 4. Atmospheric Constants. (After: U.S. Standard Atmosphere, 1976)

Initial calculations assumed the geodetic average of 6371.0 km as the radius of the Earth. As the program matured, follow-on calculations called the *oblate* subroutine to account for the Earth as an oblate spheroid. Based on the current latitude, the geodetic radius of the Earth increases when measuring from the poles to the equator. This affects the atmospheric density, temperature, and pressure. Appendix B contains the *oblate* subroutine code. The initial test case derived from forty-one selected breakpoints and

loaded into *atmosv7* as Geometric Altitude, H (km), Temperature, T_{amb} (K), Pressure, P_{mb} (mbars), and Lapse Rate, L_{bp} ($\Delta T/\Delta km$). The subroutine scans the array of breakpoints to determine whether the current altitude resides in an isothermal region of the atmosphere. Then, it proceeds to calculate temperature and pressure based on Equations (26) and (27). The significance of an isothermal region affects the temperature and pressure calculations.

$$P_{mb} = P_{mbbp} \cdot e^{\left[\frac{g \cdot M_o (H - H_{bp})}{R^* \cdot T_{bp} \cdot 1000} \right]} \quad (26)$$

$$T_{amb} = T_{bp} \quad (27)$$

where:

P_{mbbp} = Pressure at the specific break point (millibars)

T_{bp} = Temperature at the specific break point (Kelvin)

For a non-isothermal region of the atmosphere, Equations (28) and (29) apply.

$$P_{mb} = P_{mbbp} \left[\frac{T_{bp}}{T_{bp} + L_{bp} (H - H_{bp})} \right]^{\left(\frac{g \cdot M_o}{R^* \cdot L_{bp}} \right)} \quad (28)$$

$$T_{amb} = T_{bp} + L_{bp} (H - H_{bp}) \quad (29)$$

where:

L_{bp} = Lapse Rate at the specific break point ($\Delta K/\Delta km$)

H_{bp} = Geometric altitude at the specific break point (km)

At the end of the subroutine, sonic velocity, c , and atmospheric density, ρ , are calculated. Thus, altitude, temperature, pressure (converted to pascals), sonic velocity, and density are passed to the *aero1* subroutine for use in the aerodynamic calculations. As a cross check, MATLAB™ successfully generated plots of temperature and pressure

versus altitude to validate the *aero1* and *atmosv7* subroutines. The MATLAB™ plots matched the graphs depicted in *U.S. Standard Atmosphere, 1976*. **Figure 16** and **Figure 17** illustrate the MATLAB™ plots generated by the *atmosv7* atmospheric model. Sonic velocity plots similarly to the atmospheric temperature graph and atmospheric density plots similarly to the atmospheric pressure graph (U.S. Standard Atmosphere, 1976, pg. 12).

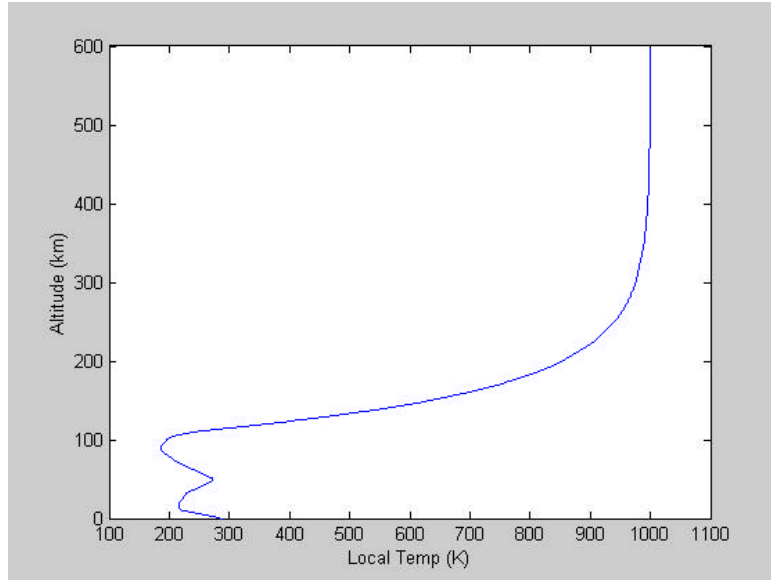


Figure 16. Temperature vs. Altitude.

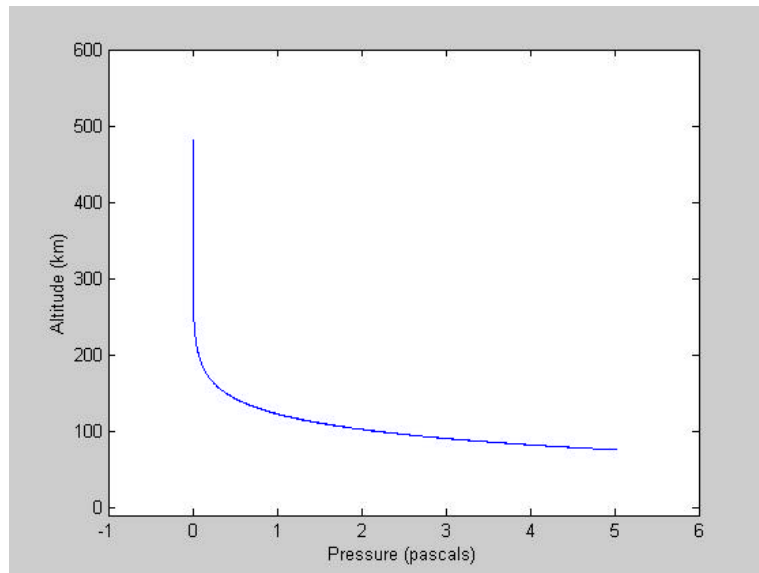


Figure 17. Pressure vs. Altitude.

F. CALCULATING THE WIND-RELATIVE TRAJECTORY

The aerodynamic subroutine, *aero1*, takes the data passed to it from the atmospheric model and calculates the parameters summarized in Equations (31) through (35). Appendix C lists the compiled *aero1* code. Since the Earth's atmosphere is rotating with respect to the inertial coordinate frame, this rotational effect must be accounted for to get high accuracy in the lift and drag calculations. With respect to the Inertial Coordinate System, the rotational velocity of the atmosphere is expressed as the inner product of the inertial position vector and the angular velocity of the Earth as shown in Equation (30).

$$V_{atmosphere} = \mathbf{v}_{earth} \times R_{inert} = \begin{bmatrix} i & j & k \\ 0 & 0 & \mathbf{w}_{earth} \\ R_x & R_y & R_z \end{bmatrix}_{inertial} \quad (30)$$

Essentially, the atmosphere travels at a slightly slower velocity than the Earth itself. Therefore, the velocity of the relative wind felt by the spacecraft varies based on its current latitude. $U_{x,y,z}$ are the respective cross products of the angular velocity vector of the Earth and the position vector of the spacecraft. U_{tot} is the magnitude of the relative wind velocity.

$$U_{tot} = \sqrt{U_x^2 + U_y^2 + U_z^2} \quad (31)$$

Equation (32) shows the relationship between the relative wind velocity and the sonic velocity to solve for the Mach number.

$$M = U_{tot} / c \quad (32)$$

Dynamic pressure results from the combination of Mach number, ambient pressure and a conversion ratio, γ , as shown in Equation (33). The conversion ratio is the same as listed in **Table 4**.

$$\bar{q} = \left(\frac{g}{2} \right) \cdot P_{mb} \cdot M^2 \quad (33)$$

Total temperature and total pressure draw from the relationship between Mach number, the conversion ratio and their respective ambient values as proven in Equations (34) and (35).

$$T_t = T_{amb} \left[1 + \left(\frac{g-1}{2} \right) \cdot M^2 \right] \quad (34)$$

$$P_t = P_{mb} \left[1 + \left(\frac{g-1}{2} \right) \cdot M^2 \right]^{\left(\frac{g}{g-1} \right)} \quad (35)$$

The data calculated in the aerodynamic subroutine was used in the analysis to calculate lift and drag on the winged reentry vehicle. Also, thrust inputs to flight path angle and angle of attack will affect the lift and drag on the vehicle as it enters the upper atmosphere. This same data could be used for follow-on research into the heating effects during re-entry, which is beyond the scope of this project.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ORBITAL REGULATOR DESIGN

A. ENERGY ANALYSIS

1. Perigee Collapse

Atmospheric drag removes energy from the spacecraft's orbital velocity and thus causes the semi-major axis, a , to shrink. If the initial orbital perigee altitude is substantially beyond the Earth's atmosphere ($>125\text{km}$), the ΔV due to atmospheric drag occurs near the orbit's perigee. The affect is that the orbit's apogee lowers rather quickly while the perigee remains relatively constant. **Figure 18** illustrates the effect of drag on an orbit's apogee. Eventually, the orbital energy becomes low enough that the orbital velocity no longer can be maintained at perigee and the orbit catastrophically collapses, which is shown in **Figure 19**. Initial test simulations run in MATLABTM showed that while the perigee remains nearly constant until the point of collapse, the instantaneous apogee altitude descends rapidly about a half an orbit ahead of the perigee collapse point. Therefore, the orbital regulator design ensures the apogee maintains an adequately high altitude such that the orbital energy remains sufficient to avoid perigee collapse.

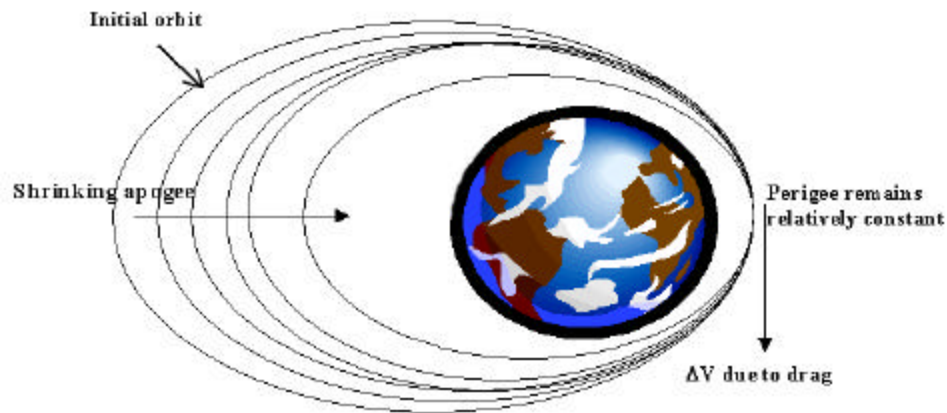


Figure 18. Effect of Drag on Apogee. (After: Whitmore Lecture Notes)

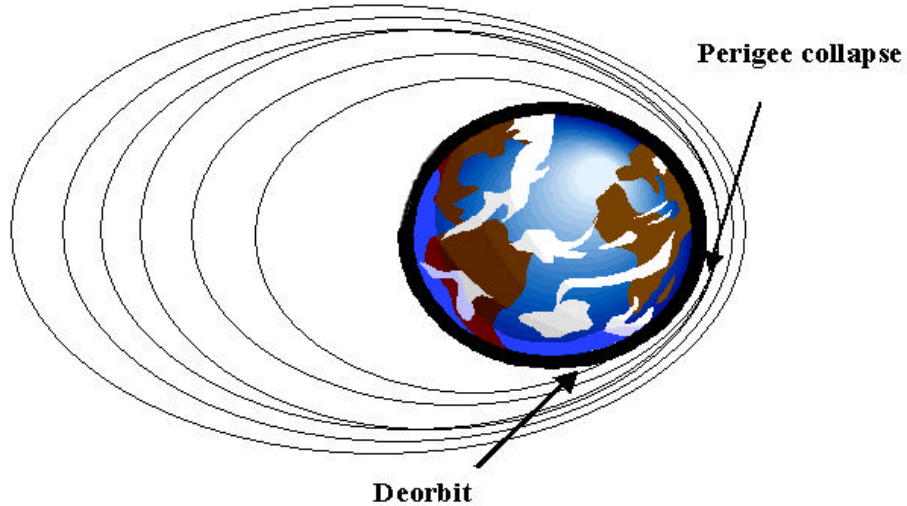


Figure 19. Perigee Collapse. (After: Whitmore Lecture Notes)

The key to maintaining a stable orbit at very low perigee altitudes is to keep the orbit apogee out of the “danger zone” in **Figure 20**, which is just above the knee in the curve. If the perigee stays above the knee, then there is enough orbital energy to maintain another full orbit. The regulator design modulates the thrust to keep the spacecraft in the “safe zone” where the perigee altitude is relatively constant.

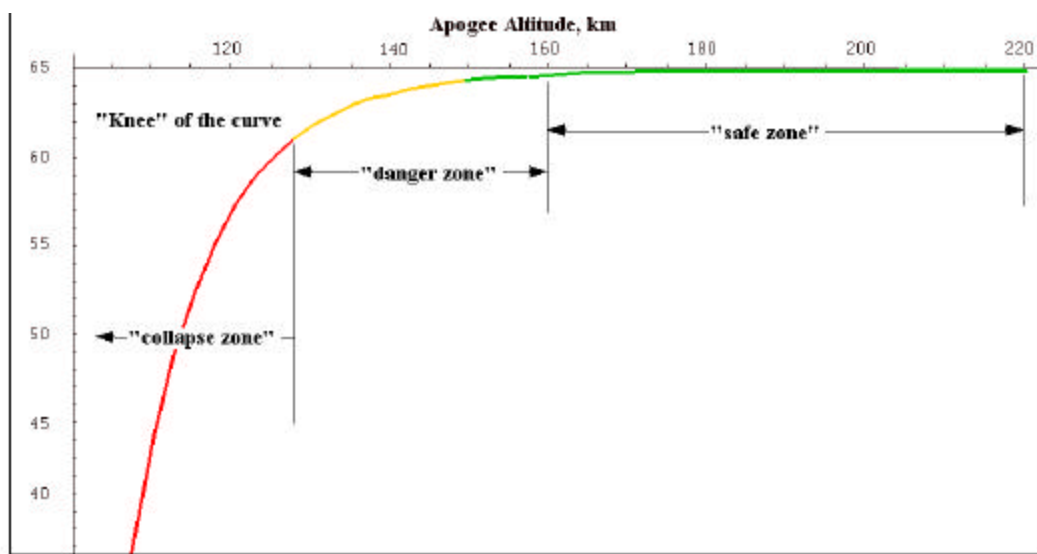


Figure 20. Apogee vs. Perigee Curve. (After: Whitmore Lecture Notes)

2. Regulator State Equation

The first step to developing the orbital regulator manifests itself from Kepler's Laws. In an ideal Keplerian situation the specific mechanical energy, \mathbf{e} , is constant as presented in Equation (36).

$$\mathbf{e}_{orbit} = -\frac{\mathbf{m}}{2a_{orbit}} \quad (36)$$

where:

μ = Gravitational parameter of the Earth ($3.986 \times 10^5 \text{ km}^3/\text{s}^2$)

a_{orbit} = Semi-major axis of the orbit (km)

If a non-conservative force is performing work on the spacecraft operating in an initial orbit, a_0 , after a period of time, denoted as \mathbf{t} , the new orbit energy level is represented by Equation (37).

$$[\mathbf{e}_{orbit}]_t = -\frac{\mathbf{m}}{2[a_{orbit}]_0} + \frac{\text{Energy added}}{m_{spacecraft}} \quad (37)$$

Where the “Energy added” term is simply the work done by the non-conservative force. Work can be represented as the integral of a force exerted over a distance in a certain amount of time as seen in Equation (38). Rearranging Equation (38) and substituting terms from Equation (36) back into Equation (37) results in Equation (39).

$$\int \overline{F}_{non-conservative} \cdot \frac{d\overline{s}}{dt} dt = \int \overline{F}_{non-conservative} \cdot \overline{V} dt \quad (38)$$

$$\frac{\mathbf{m}}{[a_{orbit}]_t} = \frac{\mathbf{m}}{[a_{orbit}]_0} - \frac{2 \int \overline{F}_{non-conservative} \cdot \overline{V}}{m_{spacecraft}} \quad (39)$$

Differentiating both sides of Equation (39), with respect to time, results in Equation (40). Rearranging Equation (40) gives the orbital decay equation as shown in Equation (41).

$$\frac{\mathbf{m}}{[a^2_{orbit}]_t} \frac{d}{dt} [a_{orbit}]_t = \frac{2 \overline{F}_{non-conservative} \cdot \overline{V}}{m_{spacecraft}} \quad (40)$$

$$\dot{a} = \left(\frac{2a^2}{\mathbf{m}} \right) \left(\frac{\overline{F}_{non-conservative} \cdot \overline{V}}{m_{spacecraft}} \right) \quad (41)$$

Remembering from the perigee collapse discussion, the objective for the orbital regulator is to modulate the thrust in such a manner that the orbital energy remains the same. For convenience of calculations, Equation (41) will be reformulated in terms of a “differential energy” term and denoted as $a - a_o$. Assuming that the regulator design is adequate such that the orbital energy at any point in the orbit is maintained close to the original orbital energy, the “differential energy” of the orbit can be approximated by the differential orbital decay rate, which is shown in Equation (42). This equation is valid only when the difference between the orbital energies is small. Otherwise, a more robust equation to account for more pronounced differences will be required at a later point in time. **Figure 21** illustrates this concept.

$$\frac{d}{dt} [a - a_o] = \frac{2[a^2 - a_o^2]}{\mathbf{m}} \left[\frac{\overline{F}_{non-conservative} \cdot \overline{V}}{m_{spacecraft}} \right] \quad (42)$$

Letting the state variable, X , equal the differential energy term, $a - a_o$, the regulator will drive the state variable to zero. Substituting this term into Equation (42):

$$\frac{d}{dt} X = \frac{2X^2}{m} \left[\frac{\bar{F}_{non-conservative} \cdot \bar{V}}{m_{spacecraft}} \right] \quad (43)$$

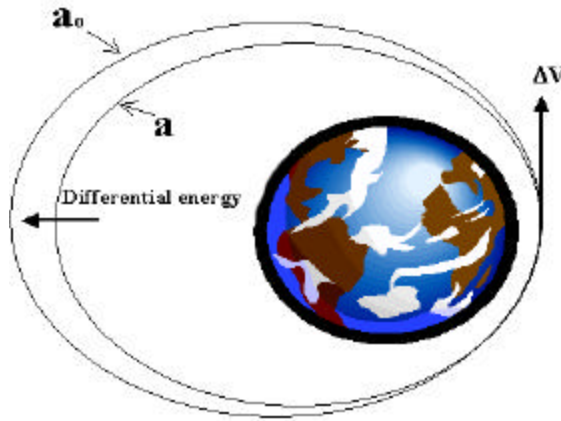


Figure 21. Differential Energy. (After: Whitmore Lecture Notes)

The next step in the regulator design procedure involves introducing a control variable. Engine throttle to control thrust returns the energy of the orbit that was removed by atmospheric drag while traveling through perigee. $F_{non-conservative}$ becomes F_{nom} of the nominal thrust from the main engine and the term, T , is multiplied in Equation (44) to account for the throttle setting.

$$\frac{d}{dt} X = \frac{2X^2}{m} \left[\frac{\bar{F}_{nom} \cdot \bar{V}}{m_{spacecraft}} \right] T \quad (44)$$

where:

$$\overline{F}_{nom} \cdot \overline{V} = (F_{nom} \sin \mathbf{q}) V_r + (F_{nom} \cos \mathbf{q}) V_n$$

Since the rocket motor is capable of only a finite range of throttle settings, constraints must be placed on T . For example, recall from **Table 1** that the AR2-3A has a throttle range from 10% to 110% and would have the control constraints listed in Equation (45).

$$\begin{aligned} T_{\min} &\leq T \leq T_{\max} \\ 0.1 &\leq T \leq 1.10 \end{aligned} \tag{45}$$

Thus, the final constrained control equation results from the combination of Equations (44) and (45).

$$\begin{aligned} \frac{d}{dt} X &= \frac{2X^2}{\mathbf{m}} \left[\frac{\overline{F}_{nom} \cdot \overline{V}}{m_{spacecraft}} \right] T \\ T_{\min} &\leq T \leq T_{\max} \\ X &= a - a_o \end{aligned} \tag{46}$$

B. CONSTRAINED CONTROL EQUATIONS

1. Hamiltonian Functional

Now that the constrained control equation exists in a usable format, the next piece of information required is a performance measurement. This scalar “cost functional” parameter, J , measures the performance of the regulator using the X and T terms as indices shown in Equation (47). The goal of the regulator is to minimize J . Any deviation of X from zero, where X measures the current semi-major axis from the original

semi-major axis, penalizes the regulator. Any unnecessary engine throttle activity, which would waste valuable fuel, also penalizes the regulator. The terms q_1 and q_2 , act as individual gain to limit the upper and lower boundaries on the orbital decay and thrust parameters. The procedures used to develop the following terms are part of the calculus of variations method (Kirk, pg. 228).

$$J(X, T) = \int_{t_0}^t \left(\frac{q_1 X^2}{2} + \frac{q_2 T^2}{2} \right) dt \quad (47)$$

Following procedures established in variational calculus, the performance index, J , could be minimized by forming a Hamiltonian functional. The basis for optimal control theory derives from the Hamiltonian functional and its ability to identify the most efficient system of equations. In orbital mechanics, a spacecraft's radius, velocity, and flight path angle (γ_{fpa}) are periodic values while the mass, pitch, and roll change as fuel is expended and non-conservative forces act on the vehicle. Equation (48) defines the Hamiltonian functional as it will be used in this discussion by incorporating Equations (46) and (47). The variable p , is referred to as the “co-state variable” and serves as a gain function for the regulator feedback control.

$$H = \left[\frac{q_1 X^2}{2} + \frac{q_2 T^2}{2} \right] + p \left[\frac{2X^2}{\mathbf{m}} \left(\frac{\overline{F}_{nom} \cdot \overline{V}}{m_{spacecraft}} \right) \right] T \quad (48)$$

The co-state equation is the first major piece to develop in this discussion and its derivation comes about by evaluating the partial of the Hamiltonian function with respect to the orbital energy in Equation (49).

$$\dot{p} = -\frac{\partial H}{\partial X} = -\left[q_1 X + p \left(\frac{4X}{\mathbf{m}} \right) \left(\frac{\bar{F}_{nom} \cdot \bar{V}}{m_{spacecraft}} \right) T \right] \quad (49)$$

The other major piece developed in this section involves the determination of the throttle setting condition. If there were no constraints on the throttle setting in the optimal control feedback expression, Equation (48) would become the *unconstrained* control equation shown in Equation (50).

$$\frac{dH}{dT} = 0 \quad (50)$$

Evaluating this partial derivative, the unconstrained control law becomes Equation (51).

$$T_{unconstrained} = -\left(\frac{p}{q_2} \right) \left(\frac{2X^2}{\mathbf{m}} \right) \left[\frac{\bar{F}_{nom} \cdot \bar{V}}{m_{spacecraft}} \right] \quad (51)$$

However, since there are allowable constraints on the throttle control input, the optimal control equations were derived using Pontryagin's Minimum Principle. This principle states that the optimal constrained functional be must less than or equal to the optimal unconstrained functional as denoted in Equation (52). Simply, the principle states that the limited (constrained) state's values can be no greater than the free (unconstrained) state's values (Kirk, pg 232).

$$H_{constrained}^{optimal} \leq H_{unconstrained}^{optimal} \quad (52)$$

Using Equation (48) to form the *constrained* optimal Hamiltonian functional, results in Equation (53).

$$H_{constrained}^{optimal} = \frac{q_2 T_{constrained}^2}{2} + p^{optimal} \left(\frac{2(X^{optimal})^2}{\mathbf{m}} \right) \left[\frac{\bar{F}_{nom} \cdot \bar{V}}{m_{spacecraft}} \right] T_{constrained} \quad (53)$$

In an optimal state, $X^{optimal}$, which is the measure between the original orbital energy and the current orbital energy, is equal to zero. So, if $X^{optimal}$ equals zero, then Equation (53) simplifies to Equation (54).

$$H_{constrained}^{optimal} = \frac{q_2 T_{constrained}^2}{2} \quad (54)$$

Applying the same methods to the unconstrained state results in Equations (55) and (56).

$$H_{unconstrained}^{optimal} = \frac{q_2 T_{unconstrained}^2}{2} + p^{optimal} \left(\frac{2(X^{optimal})^2}{\mathbf{m}} \right) \left[\frac{\bar{F}_{nom} \cdot \bar{V}}{m_{spacecraft}} \right] T_{unconstrained} \quad (55)$$

But,

$$X^{optimal} = 0$$

Therefore,

$$H_{unconstrained}^{optimal} = \frac{q_2 T_{unconstrained}^2}{2} \quad (56)$$

Substituting Equations (54) and (56) into Equation (52) and discarding similar terms applies Pontryagin's Minimum Principle to the throttle setting.

Thus,

$$T_{constrained}^2 \leq T_{unconstrained}^2 \quad (57)$$

2. Interpretation of Constrained Optimal Control

The squared property of the throttle setting inequality expression implies the use of the absolute values of both the constrained and unconstrained conditions. However, the constrained throttle must always be greater than zero because the throttle cannot be set to a negative value. Recalling from Equation (51), if the unconstrained throttle setting is greater than or equal to zero, then the positive value of the constrained throttle setting would be used in Equation (58).

$$\pm T_{constrained} \leq \pm \left\| \left(\frac{p}{q_2} \right) \left(\frac{2X^2}{\mathbf{m}} \right) \left[\frac{\overline{F}_{nom} \cdot \overline{V}}{m_{spacecraft}} \right] \right\| \quad (58)$$

Conversely, if the unconstrained throttle setting is less than or equal to zero, then the constrained throttle setting would use the negative value. This means that the constrained state would be greater than the unconstrained state. Since this condition does not obey Pontryagin's Minimum Principle, only the maximum value of the constrained throttle setting could be achieved. Therefore, $T_{constrained}$ equals T_{max} . If the minimum *allowable* throttle value is greater than zero, the constrained throttle position is set equal to the minimum throttle position. The conditions for optimality are satisfied for this condition when the unconstrained throttle value is greater than or equal to zero. However, setting $T_{constrained}$ equal to T_{min} when the unconstrained value is between zero and the minimum allowable throttle position violates the control logic and makes the inequality expression sub-optimal. The strategy developed to optimize the control functions relies upon the initial values for the co-state, p , and the gain values, q_1 and q_2 . The *orbreg* subroutine formulates this control logic and is presented in Appendix J.

3. Summary

Recalling from above, the two pieces of key information for the orbital regulation strategy used in the *orbreg* subroutine involve Equations (49) and (51).

$$\dot{p} = -\frac{\partial H}{\partial X} = -\left[q_1 X + p \left(\frac{4X}{\mathbf{m}} \right) \left(\frac{\bar{F}_{nom} \cdot \bar{V}}{m_{spacecraft}} \right) T \right] \quad (49)$$

$$T_{unconstrained} = -\left(\frac{p}{q_2} \right) \left(\frac{2X^2}{\mathbf{m}} \right) \left[\frac{\bar{F}_{nom} \cdot \bar{V}}{m_{spacecraft}} \right] \quad (51)$$

The initial conditions for the formulation start with some initial orbit at a_o . Thus, the expression $X_o = a - a_o$ equals zero. In addition, the initial throttle position and co-state variable in Equation (49) will equal zero. Therefore, integrating over time solves for the updated co-state values. When the expression for the unconstrained throttle position is less than or equal to zero set the constrained throttle position to zero. If the expression for the unconstrained throttle position is between zero and the minimum allowable throttle position, the throttle position is constrained to the minimum value. Otherwise, the control logic would become sub-optimal and too much oscillation occurs in the throttle setting. When the expression for the unconstrained throttle position is greater than or equal to zero but less than or equal to the maximum allowable throttle position, set the throttle position equal to the unconstrained value. If the unconstrained value for the throttle position exceeds the maximum allowable setting, then the throttle is set to its maximum position. Equation (59) summarizes the concept of the orbital regulation control logic.

$$\begin{aligned} & \text{if } \dots T_{unconstrained} \geq T_{\max} \\ & \quad \rightarrow T = T_{\max} \\ & \text{elseif } \dots T_{\min} > T_{unconstrained} > T_{\max} \\ & \quad \rightarrow T = T_{unconstrained} \\ & \text{elseif } \dots 0 < T_{unconstrained} \leq T_{\min} \\ & \quad \rightarrow T = T_{\min} \\ & \text{else} \\ & \quad \rightarrow T = 0 \end{aligned} \quad (59)$$

The purpose of the co-state variable is to act as an adaptive gain function. The larger the q_1 value, the more the regulator wants to match the current orbit's semi-major axis, a , to the original semi-major axis, a_o . Essentially, q_1 raises the slope of the entire expression and thus introduces more jitter into the throttle. The larger the q_2 value, the more it costs in terms of fuel. A greater amount of fuel is consumed because of its inverse proportion relationship to the co-state adaptive gain. The crux of the analysis revolves around finding the best ratio between the gain values in order to achieve the most efficient usage of the throttle and fuel while maintaining orbital energy near a constant value.

V. SIMULATION TOOLS

A. MATLAB™/SIMULINK™

The implementation of the codes discussed in the above sections began with the SIMULINK™ package available as part of the MATLAB™ toolbox. **Figure 22** shows the entire simulation as developed in SIMULINK™. The starting point begins with the initial conditions block that is written in the particular flight subroutine. An example flight subroutine is listed in Appendix K. The initial values pass through the integrator and then are distributed throughout the wiring diagram to calculate the necessary variables. The individual subroutines are called when needed and then multiplexed into the *forces* program. The equations of motion listed in Equation (18) are de-multiplexed and then fed into the integrator in order for the process to be repeated over the total simulation time. A fixed-step, Runge-Kutta propagator with a 5 second step time was chosen for the simulation parameters. The design of the orbital regulator and the equations of motion lent themselves to this decision. A variable-step, Dormand-Prince propagator was evaluated for comparison and produced similar results. However, it required longer computation time. Therefore, the fixed-step option was exercised throughout the analytical procedure.

In addition, the simulation presents display boxes to show program input and output values at each point throughout the wiring diagram. The display box, “init conds” shows the initial conditions from the starting flight subroutine. Display box, “aero1out” shows the outputs from the *aero1* subroutine and those values compiled from the *atmosv7* subroutine. Each of the *orb1*, *orb2*, and *orb3* subroutines possess their own displays culminating in the display box labeled “orbddataout” as the final values that input into the *forces* routine. The “forcesout” display illustrates the results from the equations of motion that become the next set of values that pass through the integrator. Three graphs highlight in real time the important aspects of the simulation. An XY graph shows a top-down look of the orbit as the vehicle propagates in the orbit. Two scopes illustrate the perigee and apogee altitudes and the throttle positions, respectively. Various outputs

from the different subroutines were saved to the workspace for later analysis and plot comparisons.

The basic procedure involved setting up the initial conditions for the particular flight characteristic studied. The simulation parameters were then checked for accuracy: fixed-step, Runge-Kutta propagator with a 5 second step size, and total simulation time. Referred to as the stop time, the total simulation time depended on the characteristics of the initial conditions. The first several flights were given stop times of 2.0×10^5 seconds, which represents about 37 orbits. Approximately 30 to 40 minutes of computation time were required for these initial test simulations. That stop time was chosen as a tradeoff between the amount of computation time required and the need to gather data out to near fuel exhaustion for the vehicle. Once the variables were refined over several iterations, shorter stop times were used to shorten the computation time. If a simulation time extended beyond the fuel capacity of the vehicle, a warning message appeared but the simulation continued until its designated stop time for completeness. Once finished, the data was plotted from a separate plot file (for that particular run) and only data prior to fuel exhaustion was examined.

B. LABVIEW™

A product by National Instruments™ Inc., LABVIEW™ offers another approach to real-time simulation. This tool was used in a similar fashion as the SIMULINK™ model. The front panel display acts as the set up for the initial conditions while the wiring diagram exists in the background to run the program. Initial conditions are set in the leftmost boxes (top three) while output values appear in the remaining boxes. Graphical representations illustrate a top-down look as well as an equatorial view of the vehicle as it propagates in its orbit. Real-time plots show respectively perigee and semi-major axis, dynamic pressure vs. time, and latitude over a mercator projection of the Earth on three separate graphs. The entire display is too large to display as a concise picture in this report, but a snapshot of the center panel is shown in **Figure 23**.

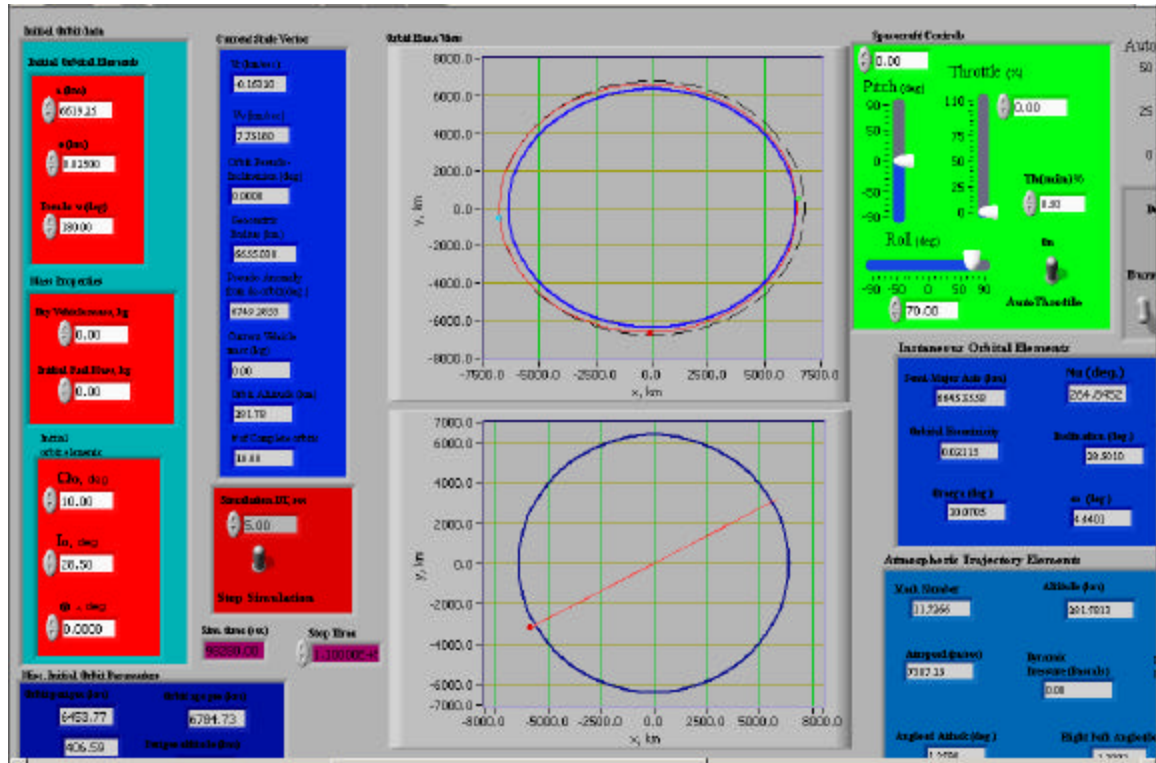


Figure 23. LABVIEW™ Center Panel.

A fixed-step, trapezoidal rule integrator resulted in the most efficient propagator for this simulation tool. A five second time step was used for consistency with the SIMULINK™ model. The stop time was set arbitrarily high to ensure the simulation would run until the vehicle exhausted its fuel. A warning message would appear when the vehicle ran out of fuel and the simulation stopped. Data was logged to that point and then saved under a new name so that the next simulation started under its new set of initial conditions.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. DISCUSSION OF RESULTS

A. ANALYSIS OF THE AERODYNAMIC LIFT VECTOR

1. Tuning the Orbital Regulator

The first step to begin the analytical process required the tuning of the orbital regulator. This involves the selection of the variables q_1 and q_2 in Equations (49) and (51). Recalling from these equations, q_1 is a control gain variable with the units of $1/\text{km}^2\text{sec}$ and q_2 has the units of $1/\text{sec}$. Their units derive from the equation balance so their individual absolute values do not pose noteworthy significance. However, taken together as a ratio, they weigh heavily in the outcome of the performance of the regulator. The gain variable, q_1 , controls the difference between the original and current orbital semi-major axis, which is directly proportional to the orbital energy. The larger the value of q_1 , the greater the regulator wants to match the current orbital energy to the original orbital energy. Thus, more throttle oscillations occur. With its inverse proportionality, a larger gain variable, q_2 , introduces more fuel usage into each throttle input. The first portion of the analysis explored the ratio of q_2 to q_1 , known as Q , and its effects on the maximum inclination change of the vehicle.

The initial flight simulations sought to determine the most efficient gain variable ratio for the maximum amount of inclination change. The simulations were timed to take the vehicle to near or full fuel exhaustion. This removed the variability of excess mass while the vehicle propagated in its orbit. For the cases where the fuel was not fully exhausted, the orbits were nearly circularized by the maneuvers and further plane change would not have been extracted. **Table 5** lists the various trials adjusting Q from low to high ratios. The initial conditions set the starting semi-major axis at 6619.25 km with an eccentricity of 0.028. The pitch and roll were set at values near the maximum lift-to-drag ratio (L/D) for the most efficient performance. Derived from open source information, a series of tests were conducted to establish the probable maximum L/D for an X-37 based vehicle. **Figure 24** illustrates the results of these tests measuring the lift-to-drag ratio versus the angle of attack, α . The scales are normalized to the maximum values on the

graph to avoid any possible technology transfer or proprietary issues concerning these specific tests. The basis for these tests helps validate that a winged re-entry vehicle operating in both a space and atmospheric environment would perform with peak efficiency at or near maximum L/D. All follow-on flight simulation results are based on analytical procedures that do not intend to uncover proprietary information.

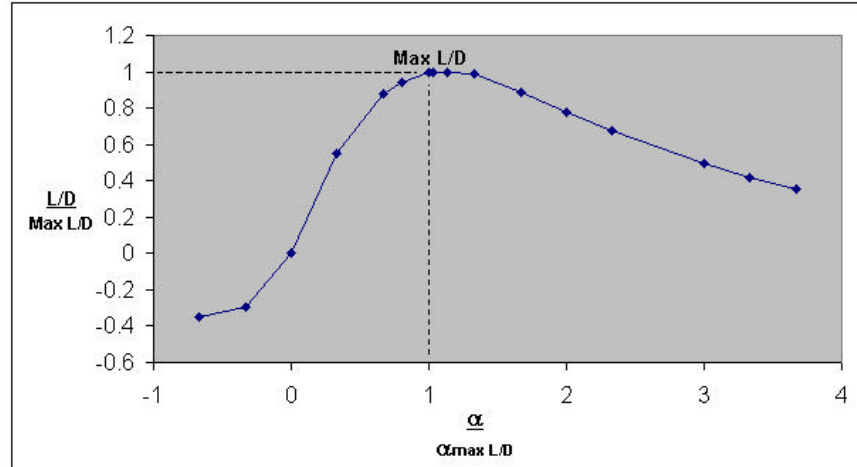


Figure 24. Normalized Max L/D vs. Angle of Attack.

Although the initial conditions later proved near optimal, follow-on tests showed that the overall outcome for Q remained the same. A Q ratio of 333.33 produced the best inclination change at maximum fuel consumption. Therefore, all follow-on flight simulations were conducted with $q_1 = 0.15$ and $q_2 = 50.0$ for a ratio of 333.33. **Figure 25** graphically represents the results of the test simulations. It is important to note that changing the magnitudes of the gain variables while keeping the ratio the same did not affect the performance of the regulator.

Flight	Gain Ratio	Δi (deg)	Fuel Consumed (kg)	
8a	83.33	14.849	1261.75	exhaustion
8b	166.67	14.859	1261.75	exhaustion
8c	333.33	14.921	1261.75	exhaustion
8h	350.00	14.850	1261.75	exhaustion
8g	400.00	14.780	1255.60	near exhaustion
8d	666.67	14.258	1216.00	near circular orbit
8e	1333.33	10.055	893.70	near circular orbit
8f	3333.33	6.146	568.70	near circular orbit

Table 5. Gain Ratio Results.

Flights 8d-e-f produced near circular orbits because of the gain value selection, where q_1 was low and q_2 was high. This means that deviations from the original orbit are “inexpensive” (low q_1 value) and fuel usage is “expensive” (high q_2 value). The regulator allows the perigee to climb until eventually the spacecraft altitude is so high that no more inclination change can be extracted, even though fuel remains aboard the vehicle. This condition is brought about since only the semi-major axis is fed back (not the perigee altitude) in the orbital regulation scheme. This is an interesting point for further study in follow-on research.

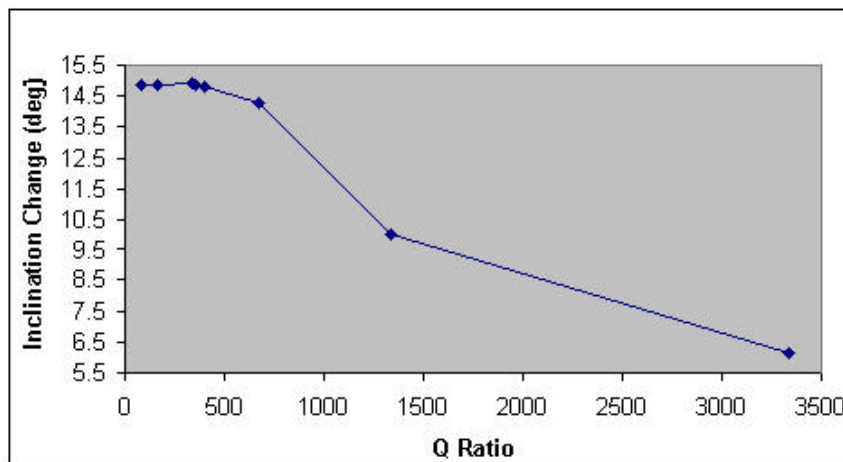


Figure 25. Q Ratio vs. Inclination Change.

2. Attitude Determination

The next phase in the analysis tested the effects of changing control inputs to the vehicle's plane change performance. This involved an iterative process of altering a single control input, such as pitch or roll, then measuring its affect on maximum inclination change. Both the SIMULINK™ and LABVIEW™ models performed simulations concurrently. Initial conditions were similar to those established previously: $a_o = 6619.25$ km, $e_o = 0.025$, $Q = 333.33$, $i_o = 28.5^\circ$, Right Ascension of the Ascending Node (RAAN) = 10.0° , and Argument of Perigee = 0.0° . Run 1 started with an arbitrary value for roll and tested several incremental values of pitch. Data gathered for analysis concentrated on the amount of inclination change for a given amount of fuel over a specified amount of time. Run 2 took the best value of pitch from Run 1 and tested several incremental values of roll. The same data was gathered and plotted for comparison. Run 3 consisted of using the best roll value from Run 2 and test several values of pitch to validate and further refine the results from Runs 1 and 2. Runs 4 and 5 altered the initial semi-major axis and eccentricity respectively, using the best results from Runs 2 and 3 and measured inclination change performance against the previous tests. The tabulated data for the SIMULINK™ and LABVIEW™ flight simulations are listed in Appendices L and M.

With an initial roll angle set at 60.0° , the results from the SIMULINK™ Run 1 indicated the best pitch angle at a value slightly greater than maximum L/D that resulted in an inclination change of 14.084° . **Figure 26** plots the results from Run 1. The same set of initial conditions were reproduced in the LABVIEW™ simulation and tested. **Figure 27** illustrates that a pitch angle several degrees larger than expected produces the best inclination change at 12.079° for the LABVIEW™ Run 1. This can be attributed to the different propagator used in LABVIEW™ for the initial conditions established. It will be shown that as the simulations become more refined, the optimal pitch reaches a value closer to that at maximum L/D demonstrated in **Figure 24**. Recall that pitch values were normalized against angle of attack at maximum L/D to reserve proprietary rights for the Boeing Company wind tunnel test data.

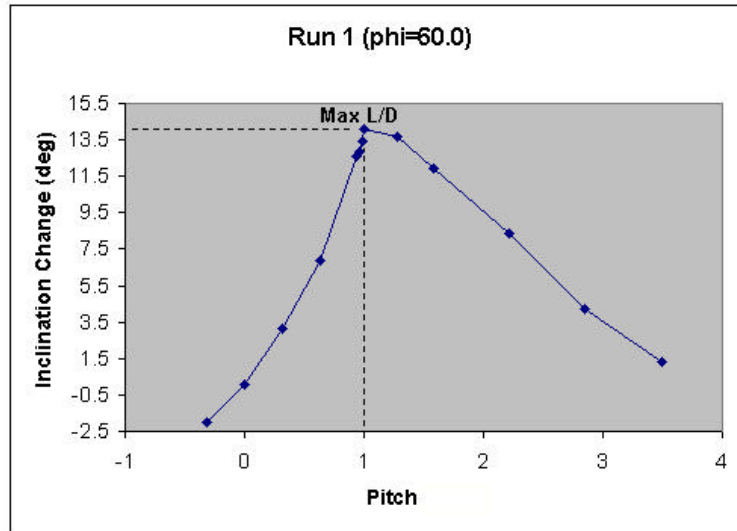


Figure 26. SIMULINK™ Run 1.

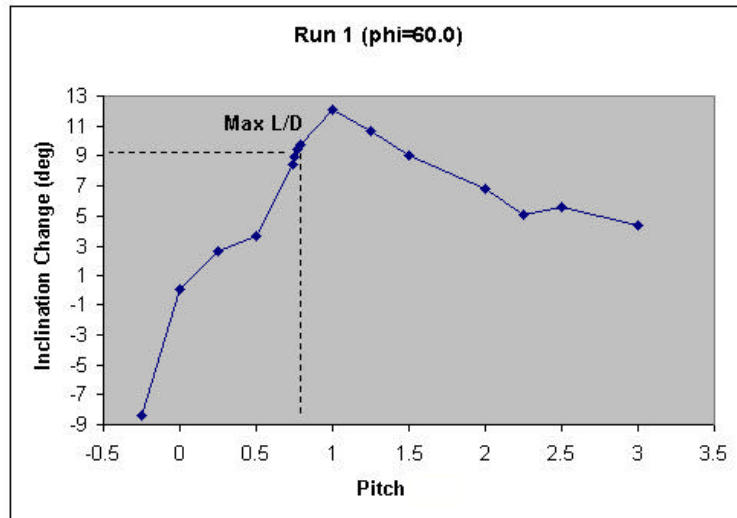


Figure 27. LABVIEW™ Run 1.

Using their respective pitch angles in the next series of simulations, Run 2 measured various roll angles and the comparison plots between the SIMULINK™ and LABVIEW™ models are shown in **Figure 28**.

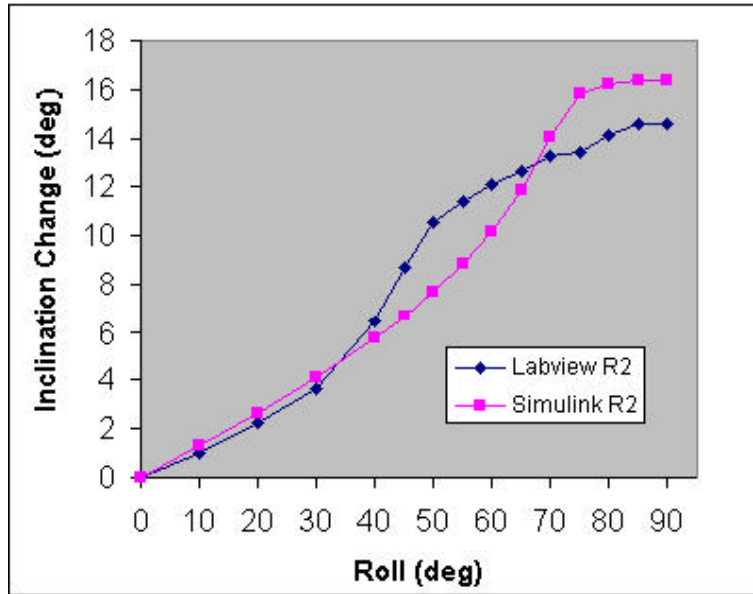


Figure 28. Comparison between Run 2 Simulations.

From **Figure 28**, it can be seen that both simulation programs produced the maximum inclination change at a high degree of roll. The SIMULINK™ model resulted in a 16.404° plane change while the LABVIEW™ model resulted in a 14.585° plane change. Both of these results occurred at complete fuel usage with a roll angle of 85.0°.

Run 3 for both simulation programs took the roll angle from Run 2 and tested various pitch angles to validate the results from Runs 1 and 2. **Figure 29** illustrates the comparison plots between the simulation programs for Run 3. From this figure, the MATLAB™ simulation produced a 16.428° plane change at maximum L/D and maximum fuel consumption. LABVIEW™ showed a 14.847° plane change at a pitch angle a few tenths of a degree less than maximum L/D and maximum fuel consumption. The shapes of the curves resemble those from Run 1 for both simulations. The results from Run 3 are more refined and the different mathematical propagators explain the deviation between the two simulation programs. Also of note, the higher the pitch and roll angles, the faster the vehicle achieved its maximum inclination change and fuel consumption. Initial flight simulations required 20 or more orbits before fuel exhaustion or maximum plane change. As the tests approached the optimal pitch or roll angle, fewer orbits were required before fuel exhaustion (8 – 10 orbits).

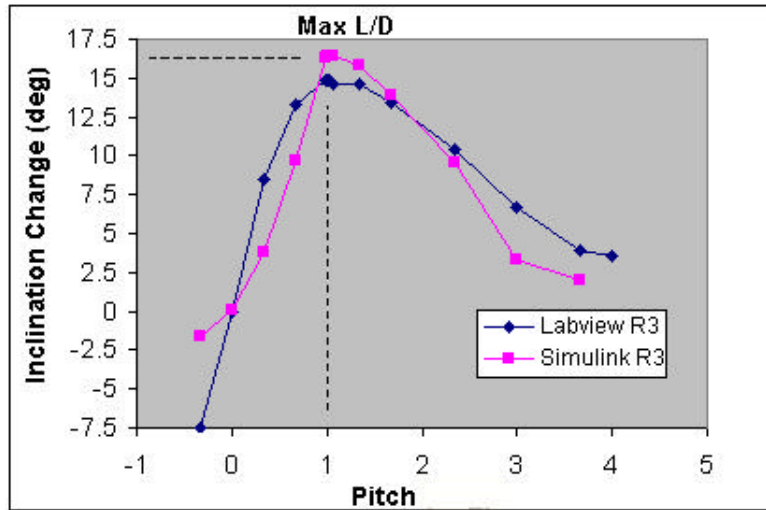


Figure 29. Comparison between Run 3 Simulations.

Follow-on simulations explored the effect of changing the original semi-major axis and eccentricity. Run 4 altered a_o to 6600.0km, which lowered the starting perigee and apogee altitudes. Recall from **Figure 20** that as long as the apogee altitude remains above 165 km with a lowering perigee (~64 km) the vehicle will stay in the “safe zone” and perigee collapse will be avoided. For Run 4, perigee altitude, H_p , equaled 64.0 km, apogee altitude, H_a , equaled 394.0 km, and e_o remained the same at 0.025. Using the best pitch angle from their respective Run 3’s, each simulation tested a short series of roll angles. The compiled data tables are presented in Appendices L and M. Run 5 made the eccentricity steeper at 0.028. The semi-major axis was raised slightly to 6617.0 km in order heighten the perigee and apogee altitudes. An average value between Runs 1 and 3 was used for the pitch angle in Run 5 to expand the comparison parameters. **Table 6** and **Figure 30** illustrate the results from the compiled Runs 2, 4 and 5 from both simulations. The SIMULINK™ tests are labeled “SR” while the LABVIEW™ tests are labeled “LR.”

On the surface, both simulations point to a roll angle of 85.0° as the producer of the maximum inclination change. However, LABVIEW™ simulations with roll angles higher than 75.0° collapsed within the first orbit. Therefore, their results were disregarded. The SIMULINK™ tests did not share the problem of orbital collapse, although the perigee altitudes did lower into a considerably precarious regime, one that is

unlikely supportable by a real flight vehicle. A more tuned regulator with a sophisticated heating analysis package would be required to solve the situation. As a result, a roll angle of less than 75.0° is recommended to avoid a dynamically unstable situation and heating concerns. In summary, a pitch angle at maximum $L/D \pm 0.35^\circ$ and roll angle of $70.0^\circ \pm 5.0^\circ$ produced the most inclination change for a given amount of fuel.

Simulation				
Run	ao (km)	eo	Apogee (km)	Perigee (km)
2	6619.25	0.025	82.77	413.73
4	6600.00	0.025	64.00	394.00
5	6617.00	0.028	60.72	431.28
Simulink Run				
phi (deg)	2	4	5	
65	11.865	14.954	14.930	
70	14.025	15.462	15.437	
75	15.860	15.962	15.900	
80	16.205	16.294	16.206	
85	16.404	16.527	16.454	
Table values are plane change (deg)				
Labview Run				
phi (deg)	2	4	5	
65	12.660	13.292	13.186	
70	13.298	13.971	13.926	
75	13.426	14.546	14.462	
80	14.128	14.756	15.024	
85	14.585	15.850	15.315	
Table values are plane change (deg)				

Table 6. Effects of Initial Orbit on Maximum Plane Change.

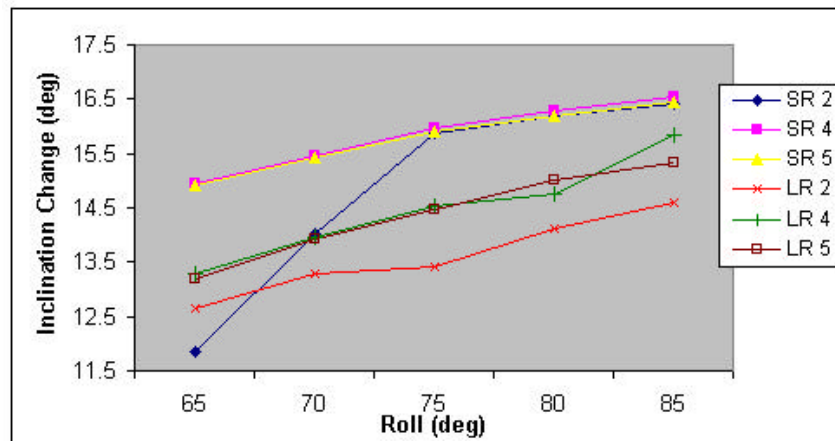


Figure 30. Comparison of Runs 2, 4, and 5.

B. DV REQUIRED FOR MAXIMUM INCLINATION CHANGE

The final phase of the research analyzed the ΔV associated with the maximum inclination change for an example flight simulation. Recalling the calculations performed for the cases in **Table 2**, the goal of that exercise was to demonstrate that a lower ΔV equated with less fuel consumption, thus cost and weight are preserved. Choosing a representative flight from the SIMULINK™ model, a detailed analysis follows to prove the fuel savings of an X-37 based vehicle. The initial conditions are listed in **Table 7**. The objective of this analysis compared the use of the described orbital transfer technique to an equivalent maneuver using a simple plane change. The first step used **Figure 31** to determine the amount of inclination change extracted given an amount of fuel loading. Conserving 200kg of fuel and oxidizer for reserve propellant and the de-orbit burn, left 1061kg of fuel expended for the orbital transfer. From the figure, 12.715° of plane change was achieved at that fuel load.

Parameter	Value
a_o (km)	6600.0
e_o	0.025
Pitch (deg)	Max L/D
Roll (deg)	70.0
Perigee alt (km)	64.0
Apogee alt (km)	394.0

Table 7. Example Simulation Initial Conditions.

The next several steps outline the computation of the propellant mass required for an equivalent simple plane change. Using **Figure 32**, the time at which that inclination change occurred resulted in approximately 42,900 seconds into the orbital propagation. Tracing that time on **Figure 33** determined the altitude, R_c (6589.5km), from which the orbital velocity, V_c , is computed in Equation (60).

$$V_c = \sqrt{\frac{\mu}{R_c}} \quad (60)$$

where:

μ = gravitational parameter of the Earth, $3.986 \times 10^5 \text{ km}^3/\text{sec}^2$

The calculated result from Equation (60) is 7.778 km/sec.

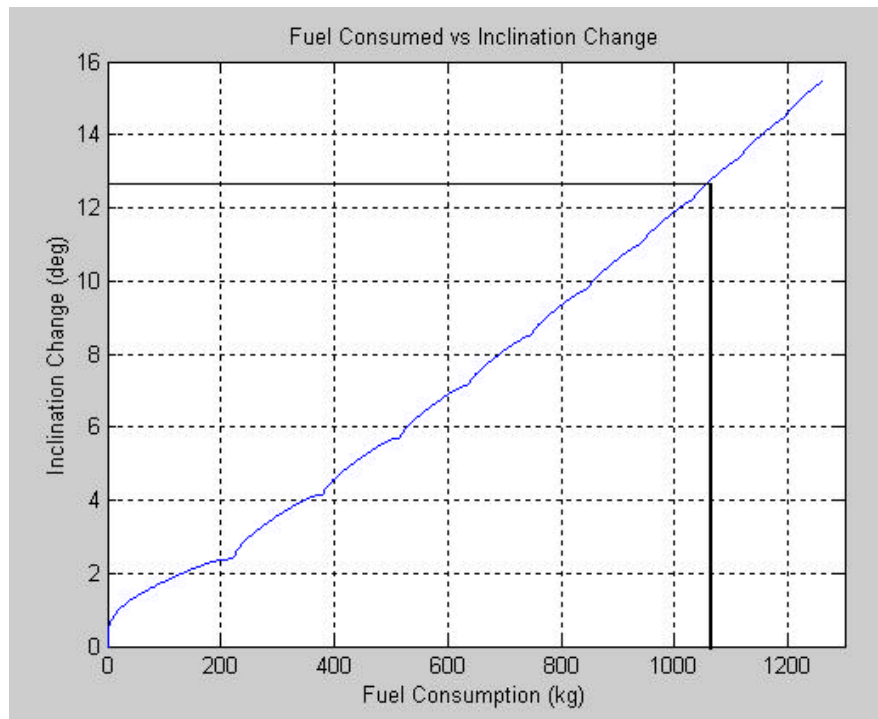


Figure 31. Fuel Load vs. Inclination Change.

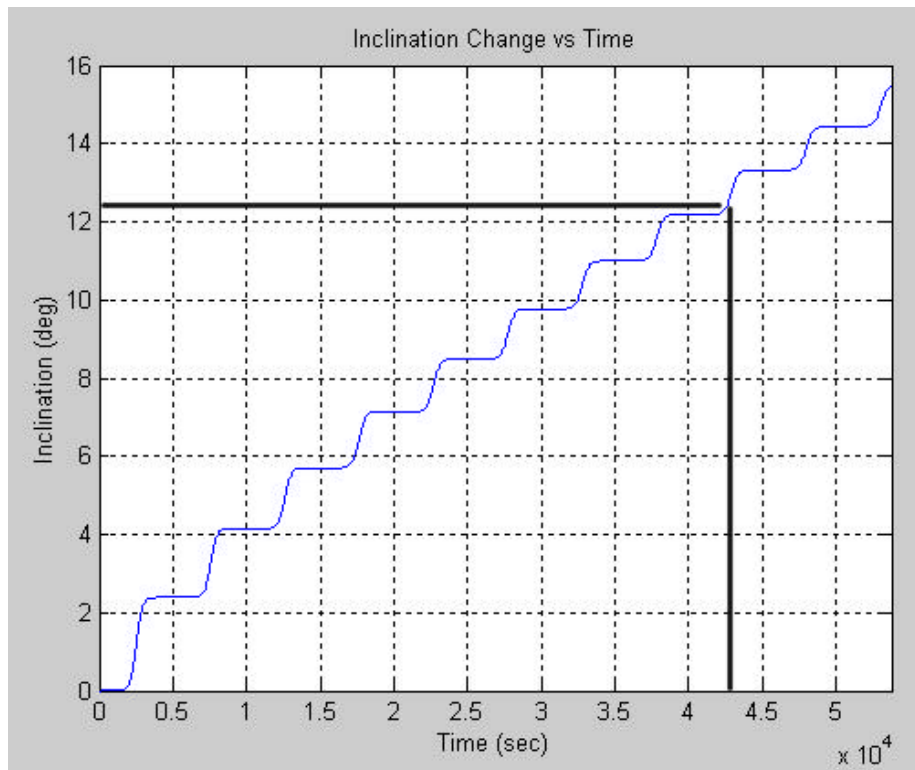


Figure 32. Inclination Change vs. Time.

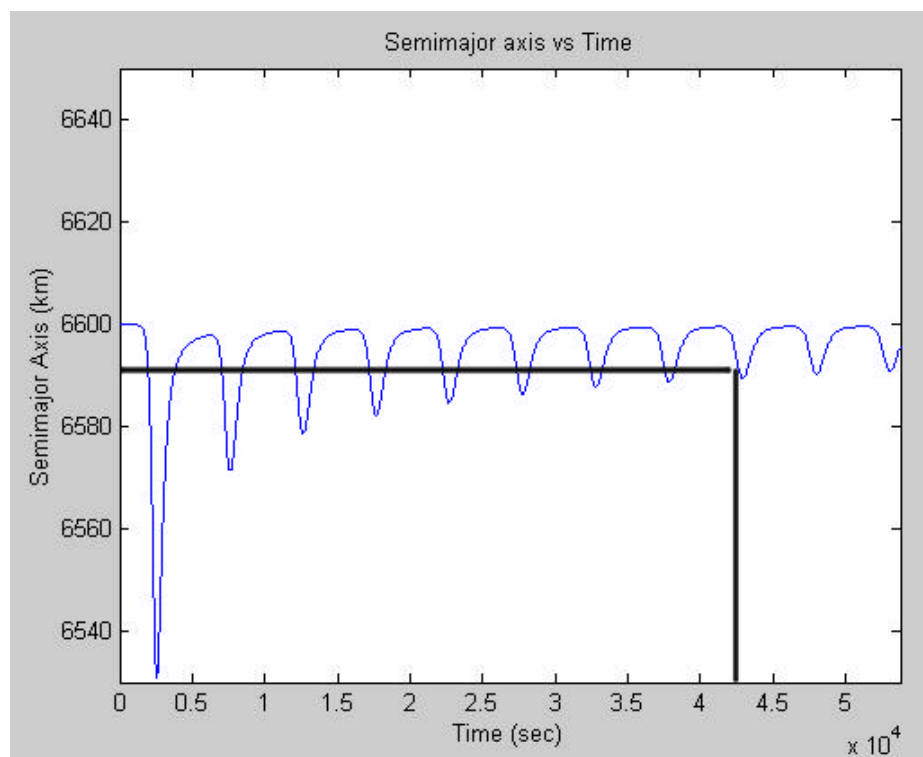


Figure 33. Altitude vs. Time.

The next step in the analysis involves the amount of mass propellant required for an equivalent simple plane change. Recall from Equation (1) that the equivalent ΔV for a simple plane change is a function of the inclination change and the orbital velocity. The result from this calculation is 1.722 km/sec.

$$\Delta V_{simple} = 2V_c \sin\left(\frac{\Delta i}{2}\right) \quad (1)$$

where:

- Δi = Inclination change, 12.715°
- V_c = Current orbital velocity at time of burn, 7.778 km/sec

The “Rocket Equation” relates the ΔV to the amount of propellant mass required for the burn as shown in Equation (61).

$$\Delta V = g_o I_{sp} \ln\left(\frac{m_{initial}}{m_{final}}\right) \quad (61)$$

where:

- ΔV = Equivalent change in velocity, 1.722 km/sec
- g_o = Acceleration due to gravity (9.806 m/s²)
- I_{sp} = Specific impulse of the rocket motor from **Table 1** (240.8 sec)
- $m_{initial}$ = Initial mass of the vehicle (4496 kg)
- m_{final} = Final mass of the vehicle (kg)

Now, $m_{initial}$ can be broken down to the vehicle’s dry mass, payload mass (including the reserve propellant), and the fuel plus oxidizer mass. The m_{final} is simply the initial mass minus the propellant mass. This relationship is shown in Equation (62)

and then rearranging the terms results in Equation (63). Collectively, the ratio of the propellant mass to the dry and payload masses is called the Propellant Mass Fraction, P_{mf} . Relating this back to Equation (62), the ratio of the initial mass to the final mass results in Equation (64). Therefore, substituting Equation (64) into Equation (61) reduces to Equation (65).

$$\frac{m_{initial}}{m_{final}} = \frac{m_{dry} + m_{payload} + m_{fuel+oxidizer}}{m_{dry} + m_{payload}} \quad (62)$$

$$\frac{m_{initial}}{m_{final}} = 1 + \frac{m_{fuel+oxidizer}}{m_{dry} + m_{payload}} \quad (63)$$

$$\Rightarrow P_{mf} = \frac{m_{fuel+oxidizer}}{m_{dry} + m_{payload}}$$

$$\frac{m_{initial}}{m_{final}} = 1 + P_{mf} \quad (64)$$

$$\Delta V = g_o I_{sp} \ln(1 + P_{mf}) \quad (65)$$

To solve for the propellant mass expended in the simple plane change starts with rearranging Equation (65) to get Equation (66). Then, substitute mass terms for P_{mf} in Equation (63), and solve for $m_{fuel+oxidizer}$ in Equation (67).

$$P_{mf} = e^{\left(\frac{\Delta V}{g_o I_{sp}}\right)} - 1 \quad (66)$$

$$m_{fuel+oxidizer} = (m_{dry} + m_{payload}) \left[e^{\left(\frac{\Delta V}{g_0 I_{sp}} \right)} - 1 \right] \quad (67)$$

Assuming the dry mass and the payload mass of the spacecraft together weigh 3500kg, the resultant of Equation (67) equals 3757.4kg. Therefore, to complete a simple plane change of 12.715° with a similarly designed spacecraft would require 3757.4kg of propellant. Recall that using the orbital transfer capabilities of an X-37 based vehicle's aerodynamic lift vector and an orbital regulator requires only 1061kg for the same 12.715° inclination change. This equates to a fuel savings of 2696.4kg. **Figure 34** illustrates that the ΔV lowers over time. Thus, in approximately 8 orbits the X-37 based spacecraft achieved its maximum inclination change using 2700kg less fuel than a conventional plane change. Appendix N displays other graphical representations, such as the perigee and apogee altitude maintenance, throttling characteristics, dynamic pressure felt by the spacecraft, and the orbital trace over the Earth. The advantages of this type of characteristic maneuver present themselves as a weight and thus, cost savings.

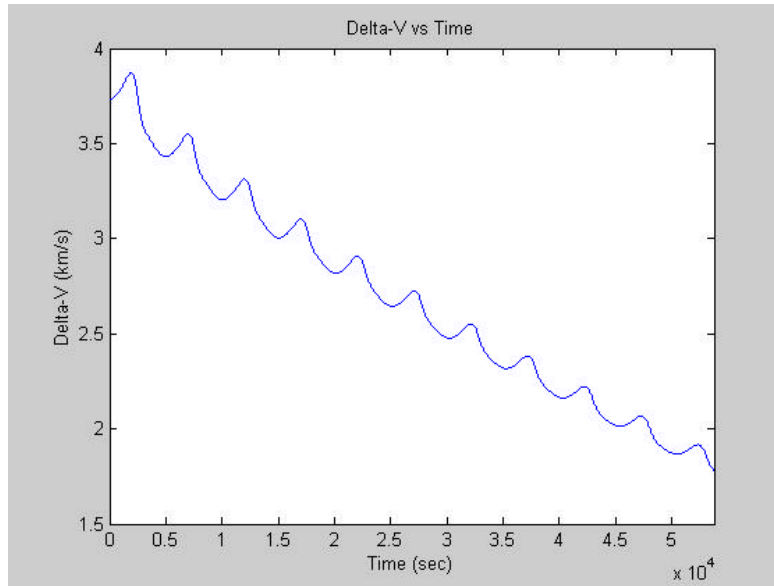


Figure 34. Delta-V vs. Time.

VII. CONCLUSION

The goals of this research succeeded in showing that the aerodynamic lift vector of a winged re-entry vehicle combined with an orbital regulator achieve upwards of a twelve-degree inclination change. The use of an X-37 based spacecraft's aerodynamic properties to conduct orbital transfers with the minimum cost in ΔV and fuel looks promising. This concept has never been published in open literature and is surely viable for further research. Between both simulation codes, the maximum plane change seen in the flight simulations was 16.5° at full fuel exhaustion. Obviously, it is unrealistic to consume every molecule of propellant, but even with a reserve set, the plane change capabilities of an X-37 based vehicle produces better results (~254% increase) over a simple plane change.

The orbital regulator program proved that it was capable of maintaining the orbital altitude to prevent perigee collapse. The best results occurred when operating at maximum L/D for the spacecraft and high roll angles ($>75.0^\circ$). However, the number of orbits required to extract maximum inclination change decreased as the roll angle increased. In the SIMULINK™ model, the perigee altitude decreased to an unrealistic value that would have caused the spacecraft to de-orbit. In the LABVIEW™ simulations, this occurred at any roll angle higher than 80.0° and the spacecraft de-orbited. Therefore, a realistic starting roll angle should be limited to less than $70.0^\circ \pm 5.0^\circ$. In addition, the roll angle becomes a concern when analyzing the heating on the thermal protection shields during the re-entry maneuvers. In a real flight vehicle, the roll and pitch could be adjusted in real time or near real time to avoid unnecessary heating on the thermal protection system.

The SIMULINK™ and LABVIEW™ simulation tools proved their utility. The LABVIEW™ program provided relatively simple user interfaces and fast computation time. SIMULINK™ demonstrated a need for longer computing time but was the driving software for batch computations and data sets. With just a limited background in programming skills, this student used the MATLAB™ tools for a design sounding board. The learning curve was exponential. The ability to log data to the MATLAB™

workspace proved to be invaluable for generating comparison plots of multiple simulation flights. Having stated that, both simulation programs have room for expansion and improvement. The next step would involve applying an optimization code to the orbital regulator routines as well as using different propagators for the integration scheme. However, both simulation programs worked extremely well as starting points and were successful in cross validation of the basic principles of the flight mechanics involved.

By the completion of this research, several follow-on projects presented themselves for further study. First, a detailed heating analysis to explore the affect of atmospheric drag on the thermal protection system would have to be conducted. This involves analyzing the temperature of the surface of the vehicle at high Mach numbers and the effect of dynamic pressure on the flight surfaces due to the pitch and roll angles. Already noted, an optimization code applied to the orbital regulator to make the throttling characteristics more efficient. Both the optimization and the heating analysis were beyond the scope of this research. Second, J_2 perturbations were neglected in this analysis because of the short amount of time in which the spacecraft performed its maneuvers. For completeness, J_2 perturbations and other external uncertainties should be added to reflect real space environment issues. Third, another mission use could be adapted to fit this type of orbital transfer technique for an X-37 based spacecraft. This involves launching into a polar orbit and using the aerodynamic lift capabilities to change the RAAN instead of the inclination. Essentially, this would spin the orbit about the poles and allow the vehicle to change orbits conceivably by ten degrees. This may have military applications for a sensor package to study targets in field of views at distances greater than its original orbit allowed. Lastly, perform an assessment of aerodynamic forces on vehicle stability during maneuvers. Rarefied flow effects during the atmospheric interface will affect the attitude control and the optimum flight path.

The lifting body projects of yesterday manifest their legacy in today's research for new space transportation and space flight vehicles. NASA and the Department of Defense have expressed the desire to expand orbital operations flexibility. Many proposals have been put forward but the X-37 Orbital Maneuvering Vehicle promises to be an excellent testbed for technology demonstration. This research project has shown

that an X-37 based vehicle offers the potential for dual use. Since the spacecraft uses aerodynamic lift to change the orbital inclination by approximately 12° , while still reserving enough fuel for a safe return from orbit, it is conceivable that mission planning can exploit this expanded operational capability to complete two mission objectives for the price of a single launch. Based on the current nominal launch cost of \$5,000 per-kilogram of payload, the resulting savings from this expanded operational flexibility has the potential to exceed hundreds of millions of dollars per year. The cost and weight savings benefit the need to expand orbital operations flexibility. The demonstration of maximum inclination change with minimum fuel consumption proves the advantages of exploring this topic further; thus expanding the boundaries of knowledge and understanding of orbital transfer techniques.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Reed, R. Dale with Darlene Lister. "Wingless Flight: The Lifting Body Story." NASA History Series. NASA Document SP-4220. Washington, D.C. 1997.
2. Dumoulin, Jim. "Shuttle Orbiter Enterprise (OV-101)." NASA Resource Library. 18 MAR 94.
<<http://science.ksc.nasa.gov/shuttle/resources/orbiters/enterprise.html>> [16 SEP 02].
3. Larson, George C. "What to Expect from the Latest Flock of X-Planes." The NeXt Generation. No Post Date Given.
<<http://www.airspacemag.com/ASM/Mag/Index/2000/DJ/next.html/>> [08 APR 02].
4. Gonzales, Daniel, Mel Eisman, Calvin Shipbaugh, Timothy Bonds, and Anh Tuan Le. Proceedings of the RAND Project AIR FORCE Workshop on Transatmospheric Vehicles. Santa Monica, CA: RAND, 1997.
5. Pike, John. "X-38." World Space Guide. 25 DEC 98.
<<http://www.fas.org/spp/guide/usa/launch/x-38.htm/>> [11 JUL 02].
6. Pike, John. "Military Spaceplane X-40 Space Maneuver Vehicle (SMV) Integrated Tech Testbed." Military Space Programs. 14 JAN 99.
<<http://www.fas.org/spp/military/program/launch/msp.htm/>> [23 APR 02].
7. Cast, Jim, Dom Amatore, Leslie Williams, and Erik Simonsen, "X-40A Free Flights Completed at Dryden; X-37 Next." NASA Press Release 01-98. 23 MAY 01. <<http://quest.arc.nasa.gov/aero/news/05-23-01.txt>> [23 APR 02].
8. "2000 Reusable Launch Vehicles Programs and Concepts." U.S. Government Programs. Associate Administrator for Commercial Space Transportation (AST). JAN 2000.
<<http://www.fas.org/spp/guide/usa/launch/2000rlv.pdf>> [11 JUL 02].

9. Anarde, BG USAF. "Military Spaceplane (MSP) and Reusable Launch Vehicle Study." 24 AUG 01.
<<http://www.nasawatch.com/X.33/08.24.01.mil.space.plane.ppt>> [16 JUL 02].
10. Andrews Space & Technology. "X-37 Specs." 2001.
<http://www.spaceandtech.com/spacedata/rlvs/x37_specs.shtml> [16 SEP 02].
11. Whitmore, Stephen A. "Mathematics of Applied Physics." Space Systems Academic Group, SS3900. Naval Postgraduate School, Monterey, CA. 26 JUN 01. <<http://web.nps.navy.mil/~sp3900/>> [18 JUL 02].
12. Sellers, Jerry Jon. Understanding Space: An Introduction to Astronautics, 2nd ed. New York: McGraw-Hill Companies, Inc., 2000.
13. Vallado, David A. Fundamentals of Astrodynamics and Applications, 2nd ed. El Segundo, CA: Microcosm Press and Boston, MA: Kluwer Academic Publishers, 2001.
14. Freidberger, W.F., ed. International Dictionary of Applied Mathematics, Princeton, NJ: Van Nostrand Co. Inc., 1960.
15. National Oceanographic and Atmospheric Administration, NASA, and USAF, U.S. Standard Atmosphere, 1976. Washington, DC: 1976.
16. Kirk, Donald E. Optimal Control Theory: An Introduction, New Jersey: Prentice-Hall Inc., 1970.

APPENDICES

Appendix A – *atmosv7.m*

```
% John P. Pienkowski
% Naval Postgraduate School
% September 2002

% ***** Atmosphere Subroutine *****

% Given altitude referenced from center of the Earth, this subroutine evaluates various
atmospheric parameters. Calls the "oblate" subroutine to account for the Latitude of the Earth as a oblate
spheroid. Reference 1: U.S Standard Atmosphere, 1976; NOAA, NASA, USAF; OCT 1976.

% *****

function [H,T,P,rho,c] = atmosv7(r,Lat,iload);

% Inputs:
% r...orbital radius at current position from center of Earth
% Lat...Latitude, degrees
% Outputs:
% H...local altitude in km
% P...pressure in pascals
% T...temp in deg K
% rho...density in kg/m^3
% c...sonic velocity in m/s
% Npoints... Number of test points

% Variables for use in loading of atmospheric data tables
global atmos_data;
TRUE = 1;
FALSE = 0;

% ***** Beginning of Declarations *****

% "atmos_data" is a spreadsheet containing 41 break points of arbitrary altitudes from U.S.
Standard Atmosphere, 1976. It is a field imported to Matlab from Excel using the Matlab Import Wizard.
"nn" is the variable given for the length of the data field. "iload" is a term used that will call the variables
and then save them locally as stored values.

if(iload == TRUE)
load atmos_data;
atmos_data = data;
end
nn = length(atmos_data);
```

```

% *** Constants ***
% Gas constant (N*m/kmol*K) -> pg 3 of Ref.1
    Rstar = 8.31432E3;
% Molecular weight of air (kg/kmol) -> pg 9 para 1.2.4 of Ref.1
    Mo = 28.9644;
% Gas Constant for Air (N*m/deg K)
    Rair = Rstar/Mo;
% Gravitational acceleration at sea level (m/s^2)
    gprime = 9.80665;
% Ratio between Cp/Ct (unitless)
    gamma = 1.4;
% Radius of Earth (km)
% Call Oblate Earth Subroutine:
    [Re] = oblate(Lat);
% Re = 6371.0;
% Altitude (km)
% Test case:
% Hbp = [0.0 11 20 32 47 51 71 84.5 91 100 125 140 160 180 200];
% Read as 'all variables from 2 to nn (41) in col 1 of "atmos_data".
    Hbp = atmos_data(2:nn,1);
% Temperature (K)
    Tbp = atmos_data(2:nn,2);
% Pressure (mbars)
    Pmbbp = atmos_data(2:nn,3);
% Lapse Rate (delta K/delta km)
    Lbp = atmos_data(2:nn,6);
% Number of brk points:
% Listed as "nn-1" because spreadsheet had title cell, therefore this will start it at the first data
cell.
    Nbp = nn-1;
% ***** End of Declarations *****
% ***** Beginning of Executable Code *****
% Compute Geometric Local Altitude
    H = r-Re;
% Scan the array of altitude brk points
    for j = 1:Nbp

```

```

        if H > Hbp(j)
            idx = j;
        end
    end
end

% Check to see if in an isothermal region:
% Logical "for loop" for isothermal regions 3,8, or 9
for idx = 1:Nbp
    if idx == 3
        ISO = 1;
    elseif idx == 8
        ISO = 1;
    elseif idx == 9
        ISO = 1;
    else
        ISO = 0;
    end
end

% Compute Pressure and Temperature (standard temp and press 1976)
if ISO == 1
    % ***Isothermal Region of Atmosphere***
    % Calculate Press -> Must convert H & Hbp into meters by dividing by 1000
    expn = gprime*Mo*(H-Hbp(idx))/(Rstar*Tbp(idx)*1000);
    Pmb = Pmbbp(idx)*exp(-expn); % ***Ref.1, pg 12, eqn 33b
    % Calculate Temp
    T = Tbp(idx);
else
    % ***NOT Isothermal Region of Atmosphere***
    % Calculate Press -> Must convert Lbp into meters by dividing Lbp by 1000
    expn = gprime*Mo*1000/(Rstar*Lbp(idx));
    Term1 = Tbp(idx)+Lbp(idx)*(H-Hbp(idx));
    Term2 = Tbp(idx)/Term1;
    Pmb = Pmbbp(idx)*(Term2^expn); % ***Ref.1, pg 12, eqn 33a
    % Calculate Temp
    T = Term1;
end

% Sonic Velocity (m/s)

```

```

c = sqrt(gamma*Rair*T); %***Ref.1, pg 18, eqn 50
% Converts Pressure from mbars to pascals
P = Pmb*100;
% Calculate density for given altitude (kg/m^3)
rho = P/(Rair*T); %***Ref.1, pg 15, eqn 42

```

Appendix B – *oblate.m*

```

% John P. Pienkowski
% Naval Postgraduate School
% September 2002

% ***** OBLATE SPHEROID SUBROUTINE *****

% This subroutine computes the local radius of an oblate Earth as a function of the geocentric
latitude.
% *****

function[Re] = oblate(Lat);
% Inputs:
% Lat...geocentric latitude, deg
% Outputs:
% Re...Radius of oblate Earth, km
% Constants:
rad = pi/180.0;
a = 6378.13649; % Equatorial Radius of the Earth, km
b = 6356.75170; % Polar Radius of the Earth, km

% ***** Executable Code *****

% Calculate Eccentricity of the Earth:
Ece = sqrt((a^2 - b^2)/a^2);
% Normalize Local Radius to Equatorial Radius:
ratio = sqrt((1 - Ece^2)/(1 - (Ece*cos(rad*Lat))^2));
Re = a*ratio;

```


Appendix C – *aero1.m*

% John P. Pienkowski
% Naval Postgraduate School
% September 2002

% ***** AERO1 Subroutine *****

% Given the current state values for r, Vr, Vnu, and theta this function calls the standard atmosphere routine and computes Mach number, qbar, Total Velocity, Flight Path Angle, Angle of Attack, Stagnation Temperature, & Total Pressure.

% *****

function[aero1out] = aero1(aero1input);

% Inputs:

r = aero1input(1);
Vr = aero1input(2);
Vnu = aero1input(3);
Rx = aero1input(4);
Ry = aero1input(5);
Rz = aero1input(6);
Vx = aero1input(7);
Vy = aero1input(8);
Vz = aero1input(9);
theta_deg = aero1input(10);
iload = aero1input(11);

% Constants & Conversions:

rad = pi/180.0;
TWOPI = 2.0*pi;
gamma = 1.4;
omegaE = TWOPI/86164.1; % Angular velocity of Earth in rads/sec

% Convert theta to radians:

theta = theta_deg*rad;

% Perform Aero calculations:

% Velocity of Lower Atmosphere (accounts for atmospheric slip due to

% Earth's rotation)

% Urel(General Equation) = [Vx;Vy;Vz] * [i j k; 0 0 omegaE; Rx Ry Rz]

Ux = Vx + (omegaE*Ry);

Uy = Vy - (omegaE*Rx);

Uz = Vz;

Urel = [Ux;Uy;Uz];

```

    Utot = 1000*sqrt(Ux^2 + Uy^2 + Uz^2); %converts into m/sec
% Local Latitude, degrees
    Lat = atan(Rz/(sqrt(Rx^2 + Ry^2)))/rad;
% Call Standard Atmosphere Routine:
    [H,Tamb,Pamb,rho,c] = atmosv7(r,Lat,iload);
% Mach number
    Mach = Utot/c;
% Dynamic Pressure
    qbar = (gamma/2.0)*Pamb*(Mach^2);
% Stagnation Pressure, pascals
    Po = Pamb*((1+((gamma-1)/2)*(Mach^2))^(gamma/(gamma-1)));
% Total Temperature
    ratio = 1.0 + ( (gamma-1.0)/2.0 )*(Mach^2);
    Pt = Pamb*(ratio^(gamma/(gamma-1.0)));
    Tt = Tamb*ratio;
% Stagnation Temperature, K
    To = Tamb*(1 + ((gamma-1)/2)*(Mach^2));
% Flight Path Angle and Angle of Attack
    fpa = atan(Vr/Vnu);
    alpha = theta-fpa;
% Convert to degrees:
    fpa_deg = fpa/rad;
    alpha_deg = alpha/rad;
aero1out =[H;Tamb;Pamb;rho;c;qbar;Utot;Lat;Mach;To;Po;fpa_deg;alpha_deg];

```

Appendix D – *orb1.m*

```

% John P. Pienkowski
% Naval Postgraduate School
% September 2002

```

```

% ***** ORB1 SUBROUTINE *****

```

% Given the current state vectors for r , V_r , V_{nu} , and θ , this function calls the `aero1` routine and computes the transformations for the perifocal position and velocity vectors.

```

% *****

```

```

function [orb1out] = orb1(nu_deg,i_deg,r,Vr,Vnu,iload);

```

```

% Inputs:
% nu_deg, i_deg, r, Vr, Vnu, iload (=1)
% Outputs:
% Rvect, Vvect

% ***** Executable Code *****

% Constants and Conversions:
    rad=pi/180.0;
    TWOPI=2.0*pi;
    Vi = 0;
% Convert nu_deg to radians:
    nu = nu_deg*rad;
% Convert i_deg to rads:
    i = i_deg*rad;
% Perform Transformations:
% Position
    Rx = r*cos(nu);
    Ry = r*sin(nu)*cos(i);
    Rz = r*sin(nu)*sin(i);
    Rvect = [Rx;Ry;Rz];
% Velocity
    Vtkm = sqrt(Vr^2 + Vnu^2);
    V1 = Vr*sin(nu) + Vnu*cos(nu);
    Vx = Vr*cos(nu) - Vnu*sin(nu);
    Vy = V1*cos(i) - Vi*sin(i);
    Vz = V1*sin(i) + Vi*cos(i);
    Vvect = [Vx;Vy;Vz];
    orb1out = [Rx;Ry;Rz;Vx;Vy;Vz];

```

Appendix E – *orb2.m*

```

% John P. Pienkowski
% Naval Postgraduate School
% September 2002

```

```

% ***** ORB2 SUBROUTINE *****

```

% Given the initial position and velocity vectors in the orbital plane, this subroutine will perform the transformation matrix operations back to the inertial plane.

```

% *****

function [orb2out] = orb2(x);

% Inputs:
    i0 = x(1);
    littleomega0 = x(2);
    bigomega0 = x(3);
    Rx = x(4); Ry = x(5); Rz = x(6);
    Vx = x(7); Vy = x(8); Vz = x(9);
    idir = x(10);

% Terms:
% i0...Initial inclination, degrees
% littleomega0...Initial Argument of Perigee, degrees
% bigomega0...Initial RAAN, degrees
% idir...variable to allow ability to turn off Rotation subroutine

% Outputs:
% Rinert... Inertial Position Vector
% Vinert... Inertial Velocity Vector

% ***** Executable Code *****

% Call Rotation subroutine:
    [MR] = rotate(i0,littleomega0,bigomega0,idir);

% Position Transformation to inertial plane
    Rvect = [Rx;Ry;Rz];
    Rinert = MR*Rvect;

% Velocity Transformation to inertial plane
    Vvect = [Vx;Vy;Vz];
    Vinert = MR*Vvect;

orb2out = [Rinert;Vinert];

```

Appendix F – *orb3.m*

```

% John P. Pienkowski
% Naval Postgraduate School
% September 2002

```

```

% ***** ORB3 SUBROUTINE *****

```

% Given the position and velocity vectors, this subroutine will compute all the orbital elements as well as eccentricity, specific angular momentum, and line of nodes vectors.

```

% *****

```

```

function[orb3out] = orb3(orb3input);
% Inputs:
% Rvect, Vvect, Re
% Outputs:
% Rt...magnitude of position vector
% Vtkm...magnitude of velocity vector
% atrue...semi-major axis
% Ec...magnitude of Eccentricity vector
% In...Inclination
% NU...True Anomaly in degrees
% Period...seconds
% Hm...magnitude of Specific Angular Momentum vector
% Omega...Final RAAN
% omega...Final Argument of Perigee
% Evect...Eccentricity Vector
% Hvect...Specific Angular Momentum vector
% Nvect...Lines of Nodes vector
% rp...Instantaneous Perigee radius (from Earth's center), km
% ra...Instantaneous Apogee radius (from Earth's center), km
% Hp...Instantaneous Perigee altitude, km
% ORBIT...Orbit count, integer
% Constants:
    TWOPI=2.0*pi;
    rad=pi/180.0;
    mu = 3.986004418E5; %km^3/sec^2

% ***** Executable Code *****

    Inclination = orb3input(1);
    Rx = orb3input(2);
    Ry = orb3input(3);
    Rz = orb3input(4);
    Vx = orb3input(5);
    Vy = orb3input(6);
    Vz = orb3input(7);
    Re = orb3input(8);
% Position
    Rt = sqrt(Rx^2 + Ry^2 + Rz^2);

```

```

% Total Velocity
    Vtkm = sqrt(Vx^2 + Vy^2 + Vz^2);

% Semi-major axis
    atrue = mu/((2*mu/Rt) - Vtkm^2);

% Inner Product of Position and Velocity
    RdotV = (Vx*Rx) + (Vy*Ry) + (Vz*Rz);

% Energy Ratio for Eccentricity Vector
    T1 = (Vtkm^2) - (mu/Rt);

% Eccentricity Vector
    ex = (T1*Rx - RdotV*Vx)/mu;
    ey = (T1*Ry - RdotV*Vy)/mu;
    ez = (T1*Rz - RdotV*Vz)/mu;
    Ec = sqrt(ex^2 + ey^2 + ez^2);
    Evect = [ex;ey;ez];

% Period
    Period = TWOPI*(atru^1.5)/sqrt(mu);

% *** Orbital Elements ***

% True Anomaly
    edotr = (ex*Rx) + (ey*Ry) + (ez*Rz);
    ER = sqrt((ex^2+ey^2+ez^2)*(Rx^2+Ry^2+Rz^2));
    if RdotV < 0.0
        NU = 360.0-(57.2958*acos(edotr/ER));
    else
        NU = 57.2958*acos(edotr/ER);
    end

% Specific Angular Momentum
    hx = (Ry*Vz) - (Vy*Rz);
    hy = (Rz*Vx) - (Vz*Rx);
    hz = (Rx*Vy) - (Vx*Ry);
    Hm = sqrt(hx^2 + hy^2 + hz^2);
    Hvect = [hx;hy;hz];

% Recompute Inclination
    In = acos(hz/Hm)*57.2958;

% Change in Inclination (Absolute Value)
    DI = Inclination-In;

% Line of Nodes
    nx = -hy;

```

```

ny = hx;
nz = 0;
Nvect = [nx;ny;nz];
% Right Ascension of Ascending Node
N = sqrt(nx^2 + ny^2 + nz^2);
if (ny >= 0)
    Omega = acos(nx/N)*57.2958;
else
    Omega = 360.0 - (acos(nx/N)*57.2958);
end
if (N==0)
    Omega = 0;
end
if (Omega==360.0)
    Omega=0;
end
% Argument of Perigee
ndote = (ex*nx) + (ey*ny) + (ez*nz);
if (ez >= 0)
    omega = acos(ndote/(N*Ec))*57.2958;
else
    omega = 360.0 - acos(ndote/(N*Ec))*57.2958;
end
if (N*Ec==0)
    omega = 0;
end
if (omega==360.0)
    omega=0;
end
% Orbit Perigee
rp = atrue*(1-Ec);
% Orbit Apogee
ra = atrue*(1+Ec);
% Perigee Altitude
Hp = rp - Re;
if Hp <= 0.0
    'Sorry Captain, you have crashed!'

```

```

        'End Simulation'
    end
% Apogee Altitude
    Ha = ra - Re;
% Delta-V Calculation
    DV = 2*Vtkm*sin((In*rad)/2);
orb3out = [Rt;Vtkm;atru;Ec;In;NU;Period;Hm;Omega;omega;Evect;Hvect;...
    Nvect;rp;ra;Hp;Ha;DV;DI];

```

Appendix G – *rotate.m*

```

% John P. Pienkowski
% Naval Postgraduate School
% September 2002

```

```

% *****
% Given the initial position and velocity vectors in the orbital plane, this subroutine will perform
the transformation matrix operations back to the inertial plane.
% *****

```

```

function [rotateout] = rotate(i0,littleomega0,bigomega0,idir);

```

```

% Inputs:
% i0, littleomega0, bigomega0
% idir...switch to turn rotation matrix on or off

```

```

    TRUE = 1;
    FALSE = 0;

```

```

% Outputs:
% wM...little omega rotation matrix
% IM...inclination rotation matrix
% WM...big omega rotation matrix
% MR...total output rotation matrix
% Conversion:
    rad = pi/180.0;

```

```

% ***** Executable Code *****

```

```

% Orbital plane coordinates are rotated to the Line of Nodes, where the orbital path intersects the
equatorial plane of the Earth. "littleomega" Rotation Matrix about the zaxis in the (-)omega (clockwise)
direction:

```

```

    if(idir == 1)

```



```

cosw = cos(littleomega0*rad);
sinw = sin(littleomega0*rad);
w11 = cosw;
w12 = -sinw;
w13 = 0.0;
w21 = sinw;
w22 = cosw;
w23 = 0.0;
w31 = 0.0;
w32 = 0.0;
w33 = 1.0;
wM = [w11 w12 w13
      w21 w22 w23
      w31 w32 w33];

% Next, coordinates system is rotated from the orbital plane
% to the equatorial plane of the Earth.
% Inclination Rotation Matrix about the newly transformed x-axis,
% in the (-)i (clockwise) direction:
cosI = cos(i0*rad);
sinI = sin(i0*rad);
I11 = 1.0;
I12 = 0.0;
I13 = 0.0;
I21 = 0.0;
I22 = cosI;
I23 = -sinI;
I31 = 0.0;
I32 = sinI;
I33 = cosI;
IM = [I11 I12 I13
      I21 I22 I23
      I31 I32 I33];

% Lastly, the coordinate system is rotated to position the x-axis in the direction of the vernal
equinox. Omega Rotation Matrix about the newly transformed z-axis, in the (-)Omega (clockwise)
direction:
cosW = cos(bigomega0*rad);
sinW = sin(bigomega0*rad);
W11 = cosW;

```

```

W12 = -sinW;
W13 = 0.0;
W21 = sinW;
W22 = cosW;
W23 = 0.0;
W31 = 0.0;
W32 = 0.0;
W33 = 1.0;
WM = [W11 W12 W13
       W21 W22 W23
       W31 W32 W33];
% Final Rotation Matrix
MR = WM*IM*wM;
end
rotateout = MR;

```

Appendix H - *forces.m*

```

% John P. Pienkowski
% Naval Postgraduate School
% September 2002

```

```

% ***** FORCE SUBROUTINE *****

```

% This subroutine will take the coefficients of Lift and Drag, as well as dynamic pressure (qbar) and calculate the necessary forces acting on the lifting body.

```

% *****

```

```

function[forcesout] = forces(F);

```

```

% Inputs:

```

```

Vr = F(1); Vnu = F(2);

```

```

i_deg = F(3);

```

```

r = F(4);

```

```

nu_deg = F(5);

```

```

m = F(6);

```

```

alphay2 = F(7);

```

```

Machx2 = F(8);

```

```

fpa_deg = F(9);

```

```

theta_deg = F(10);

```

```

qbar = F(11);

```

```

Rt = F(12);
phi_deg = F(13);
throt = F(14); % Throttle
q1 = F(15); % Variable Gain One, 1/km*sec^2
pco = F(16); % Co-State Variable
ao = F(17);

% Constants & Conversions:
Sref = 79.06944; % ft^2 - nominal reference area for lifting body
Aref = Sref*0.09290304; % ft^2 to m^2 conversion
rad = pi/180.0;
phi = phi_deg*rad; % Roll angle, rads
fpa = fpa_deg*rad; % flight path angle
theta = theta_deg*rad; % pitch angle
Fnom = 3300.0*4.44818; % Converts lbf to N
Isp = 240.8; % sec
Me = 5.9737E24; % kg
mu = 3.986004418E5; % km^3/sec^2
Fgrav = -1000.0*m*mu/Rt^2; % Converts kN to N

% ***** Executable Code *****

% Call LIFTDRAG2 subroutine:
[Liftdrag] = liftdrag2(alphay2,Machx2);
CL = Liftdrag(1,1);
CD = Liftdrag(2,1);

% Calculate Forces:
% Lift Force
L = qbar*Aref*CL;
% Drag Force
D = qbar*Aref*CD;
% L/D Ratio
L_D = L/D;
% Thrust
Fthrust = throt*Fnom/100.0; % N
% throt is percent throttle from 0 - 110%
% Fnom in N

% External Forces % All forces in Newtons (N)
Fr = L*cos(fpa)*cos(phi) + Fgrav - D*sin(fpa) + Fthrust*sin(theta);

```

```

Fnu = -(D*cos(fpa) + L*sin(fpa)*cos(phi)) + Fthrust*cos(theta);
Fi = -L*sin(phi);
Fext = [L;D;Fgrav;Fthrust];
% Semi-major axis
Vbar = sqrt(Vr^2 + Vnu^2); %km/sec
a = mu/((2*mu/Rt) - Vbar^2);
X = a - ao; %km
% Mass Flow Rate of AR2/3 Engine, kg/sec
gprime = 9.80665; %m/sec^2
EMdot = Fthrust/(gprime*Isp);
% Equations of Motion: conversion of meters to km
% Centrifugal acceleration + external radial acceleration
Vrdot = Vnu^2/Rt + (Fr/(1000.0*m)); %km/sec^2
% Coriolis acceleration + external in-direction acceleration
Vnudot = -(Vnu*Vr)/Rt + (Fnu/(1000.0*m)); %km/sec^2
idot = (Fi/rad)/(1000.0*m*Vnu); %deg/sec
rdot = Vr; %km/sec
nudot = (Vnu/rad)/Rt; %deg/sec
Mdot = -EMdot; %kg/sec
V = Vr*sin(theta) + Vnu*cos(theta);
pdot = -(q1*X + pco*((4*X/mu)*((Fnom/1000*V)/m)*throt)); %1/km*sec
adot = 0.0;
forcesout = [Vrdot;Vnudot;idot;rdot;nudot;Mdot;pdot;adot;L;D;Fgrav;Fthrust;L_D];

```

Appendix I – *liftdrag2.m*

```

% John P. Pienkowski
% Naval Postgraduate School
% September 2002

```

```

% *****Lift & Drag Interpolation Routine *****
% Using the "interp2" 2-D interpolation function, this routine
% will take an Excel spreadsheet of Lift and Drag Coefficients
% and interpolate the tables to determine a new value.
% *****
function [Liftdrag] = liftdrag2(alphay2,Machx2);
% Inputs:
% Machx2...Mach breakpoint across "x" axis

```

```

%   alphas2...angle of attack breakpoint across "y" axis
%   Outputs:
%   CL...Lift Coefficient interpolated from lookup table
%   CD...Drag Coefficient interpolated from lookup table

%***** Beginning of Executable Code *****

%   Loaded data from Excel spreadsheet using MATLAB Import Wizard:
    load XsubCLdata;
    load XsubCDdata;
    load CLsuper;
    load CDsuper;

%   Define boundaries of data from lookup tables:
    if Machx2 < 1.0
%   Subsonic Lift Coefficients
        nn = length(XsubCLdata);    % length of table values
        mmv = size(XsubCLdata);    % size of table (vector)
        mm = mmv(1,2);            % width of table values
        CL_bp_z = XsubCLdata(2:nn,2:mm); % "z" matrix of data by breakpts
        CL_alpha_bp_y = XsubCLdata(2:nn,1); % "y" matrix of data by brkpts
        CL_Mach_bp_x = XsubCLdata(1,2:mm); % "x" matrix of data by brkpts
%   General format: Zi = interp2(x,y,z,xi,yi), where xi & yi are individual
%   points or matrices entered into the argument list
        CL = interp2(CL_Mach_bp_x,CL_alpha_bp_y,CL_bp_z,Machx2,alphay2);

%   Subsonic Drag Coefficients
        ff = length(XsubCDdata);    % length of table values
        ddv = size(XsubCDdata);    % size of table (vector)
        dd = ddv(1,2);            % width of table values
        CD_bp_z = XsubCDdata(2:ff,2:dd); % "z" matrix of data by breakpts
        CD_alpha_bp_y = XsubCDdata(2:ff,1); % "y" matrix of data by brkpts
        CD_Mach_bp_x = XsubCDdata(1,2:dd); % "x" matrix of data by brkpts
        CD = interp2(CD_Mach_bp_x,CD_alpha_bp_y,CD_bp_z,Machx2,alphay2);
    else
%   Supersonic Lift Coefficients
        nn = length(CLsuper);    % length of table values
        mmv = size(CLsuper);    % size of table (vector)
        mm = mmv(1,2);            % width of table values
        CL_bp_z = CLsuper(2:nn,2:mm); % "z" matrix of data by breakpts

```

```

CL_alpha_bp_y = CLsuper(2:nn,1); % "y" matrix of data by brkpts
CL_Mach_bp_x = CLsuper(1,2:mm); % "x" matrix of data by brkpts
% General format: Zi = interp2(x,y,z,xi,yi)
CL = 1.5*interp2(CL_Mach_bp_x,CL_alpha_bp_y,CL_bp_z,Machx2,alphay2);
% Scaling factor of 1.5 added to the supersonic Lift Coefficient to
% better model characteristic differences between subsonic and
% supersonic data.
% Supersonic Drag Coefficients
ff = length(CDsuper); % length of table values
ddv = size(CDsuper); % size of table (vector)
dd = ddv(1,2); % width of table values
CD_bp_z = CDsuper(2:ff,2:dd); % "z" matrix of data by breakpts
CD_alpha_bp_y = CDsuper(2:ff,1); % "y" matrix of data by brkpts
CD_Mach_bp_x = CDsuper(1,2:dd); % "x" matrix of data by brkpts
CD = interp2(CD_Mach_bp_x,CD_alpha_bp_y,CD_bp_z,Machx2,alphay2);
end
Liftdrag = [CL;CD];

```

Appendix J - *orbreg.m*

```

% John P. Pienkowski
% Naval Postgraduate School
% September 2002

```

```

% ***** ORBITAL REGULATOR *****

```

% This subroutine develops the Optimal Control Strategy for the X-37 simulation. Based on the Calculus of Variations theory, this orbital regulator will provide enough thrust from the engine to boost the apogee while maintaining the perigee at a relatively constant altitude and preventing its collapse.

```

% *****

```

```

% Terms:

```

```

% a...Current semi-major axis (km)
% ao...Initial semi-major axis (km)
% X... (a - ao) (km)
% pco...Co-state variable => adaptive gain
% q1...Variable Gain One (1/km^2*sec)
% q2...Variable Gain Two (1/sec)
% mu...Gravitational Density Constant for Earth (km^3/s^2)
% Fnom...Nominal Thrust (N)

```

```

% theta...Spacecraft Pitch (rads)
% Tun...Throttle unconstrained (%)
% throt...Throttled constrained (%)
% Tmin...Minimum allowable throttle
% Tmax...Maximum allowable throttle
% m...Current Mass of vehicle (kg)
% mdry...Dry Mass of vehicle = 3235.0 kg
% mf...Mass of fuel = 1261.75 kg
% mo...Initial total starting mass = mdry + mf
% ***** BEGIN EXECUTABLE CODE *****
function[orbregout] = orbreg(orbreginput);

    Vr = orbreginput(1);
    Vnu = orbreginput(2);
    r = orbreginput(3);
    m = orbreginput(4);
    pco = orbreginput(5); pco_hold=orbreginput(6);
    ao = orbreginput(7);
    theta_deg = orbreginput(8);

% Constants:
    Fnom = 3300.0*4.44818; %Converts lbf to N
    mu = 3.986004418E5; %km^3/sec^2
    Tmin = 0; %Stated in (%)
    Tmax = 110; %Stated in (%)
    q1 = 0.15; %1/km^2*sec
    q2 = 50.0; %1/sec
% Q = q2/q1 = 333.33;
% pco = 0.01; %pco will probably stay at 1%
% Conversions:
    rad = pi/180.0;
    theta = theta_deg*rad;
% Calculate Semi-major axis:
    Vbar = sqrt(Vr^2 + Vnu^2);
    a = mu/((2*mu/r) - Vbar^2);
    X = a - ao; %km
% Establish Constrained Optimal Control State Equation:
    V = Vr*sin(theta) + Vnu*cos(theta);
    Tun = 100*(pco/q2)*((2*(X^2))/mu)*((Fnom/1000*V)/m); %km/s

```

```

    if (Tun <= 0.0)
        throt = 0.0;
    elseif (Tun <= Tmin)
        throt = Tmin;
    elseif (Tun >= Tmax)
        throt = Tmax;
    else
        throt = Tun;
    end
% Mass Properties:
mdry = 3235.0;
mf = 1261.75;
mo = mdry + mf;
% Fuel Consumption
fuel = mo - m;
if fuel >= mf
    'Sorry Captain, you are out of fuel!'
    'End Simulation'
end
orbregout =[throt;q1;q2;pco;Tmin;Tmax;fuel];

```

Appendix K – *flightXXX.m*

```

% John P. Pienkowski
% Naval Postgraduate School
% September 2002

```

```

% ***** FLIGHT SIMULATION *****

```

% This program generates a test case for the final simulation. The Initial Conditions vector acts as the starting point Follow the instructions to run the program. These test cases are consistent with Kepler's laws.

```

% *****

```

```

clear

```

```

% TERMS:

```

```

% ao...Initial Orbital altitude, km

```

```

% eo...Initial Eccentricity, degrees

```

```

% nu_deg...Initial True Anomaly, degrees

```

```

% BigOmega...Right Ascension of the Ascending Node, degrees

```



```

% Inclination...degrees
% LittleOmega...Argument of Perigee, degrees
% theta_deg...Pitch, degrees
% phi_deg...Roll, degrees
% pco0...Initial Co-State Variable
% mdry...Vehicle dry mass, kg
% mf...Initial fuel mass, kg
% mo...Initial Total Vehicle Mass, kg

% ***** INSTRUCTIONS *****

% 1) TREAT THE 'INITIAL ORBITAL ELEMENTS' AS THE FRONT PANEL FOR ANY
MODIFICATIONS. IF NEED BE, CHANGE THE MASS PROPERTIES.

% 2) CHANGE THE TITLE OF THE 'ORBDATAOUT' DATA FILE, 'THROTTLEOUT' DATA
FILE, & THE NAME OF THE SIM RUN TO SAVE THAT PARTICULAR RUN. THE SIM IS BASED
OFF OF 'X37SIM_KEPLER' BECAUSE THAT MODEL HAS BEEN VALIDATED TO OBEY
KEPLER'S LAWS.

% 3) CHANGE THE NAME OF THE PLOT FILE TO MATCH THE CORRESPONDING SIM
RUN

% 4) CHECK THE SIMULATION PARAMETERS TO ENSURE THAT THE STOP TIME,
TIME STEP, AND PROPAGATORS ARE CORRECT.

% ***** EXECUTABLE CODE *****

% Constants & Conversions:
TWOPI=2.0*pi;
rad=pi/180.0;
mu = 3.986004418E5;

% Initial Orbital Elements:
ao = 6600.00;
eo = 0.025;
nu_deg = 180.0;
BigOmega = 10.0;
Inclination = 28.5;
LittleOmega = 0.00;
theta_deg = Max L/D
phi_deg = 70.0;
pco0 = 0.0;      %Initial Co-state => Zero throttle

% Changes made to the orbital regulator "orbreg" subroutine:
%   q1 = 0.15;    % 1/km^2*sec
%   q2 = 50.0;    % 1/sec

```

```

% %    Q = q2/q1 = 333.33;
%    Tmin = 0 (instead of Tmin=1)
% Mass Properties:
    mdry = 3235.0;
    mf = 1261.75;
    mo = mdry + mf;
% Convert nuo to radians:
    nuo = nu_deg*rad;
% Compute radius (geocentric):
    ro = ao*(1- eo^2)/(1.0 + eo*cos(nuo));
% % Compute period (Kepler's third law):
%    Period = TWOPI*( ao^1.5 )/sqrt(mu);
% Initial Pseudo-inclination
    io = 0.00;
% Compute angular velocity of orbit
    c1 = 1.0 + eo*cos(nuo);
    c2 = ao*(1- eo^2);
    omega = sqrt(mu)*(c1^2)/(c2^1.5);
% Compute Velocity Components
    Vro = ro*omega*eo*sin(nuo)/c1;
    Vnuo = ro*omega;
% Initial Conditions Vector:
    icvector = [Vro;Vnuo;io;ro;nuo;mo;pco0;ao];
% Plots
sim('x37sim_flightfinal')
figure(1)
clf
plot(earthxy(:,1),earthxy(:,2),'b'); title('Orbital Plane');
legend('Earth at Center-blue')
hold
plot(x37(:,1),x37(:,2),'r')
hold off

save flightdataXXX orbdtaout tout phi_deg theta_deg
save thrustdataXXX throttleout tout
save aerodataXXX aerodataout tout
save MachdataXXX Machdataout tout

```

Appendix L – LABVIEW™ Flight Spreadsheets

		LABVIEW			
		Run 1			
Parameters					
ao (km) = 6619.25					
eo = 0.025					
phi (deg) = 60		Arbitrary starting point			
Dt (sec) = 5		fixed step			
		Normalized			
Flight	Total Time (sec)	theta	Δi (deg)	Fuel (kg)	
1	2.735E+03	-0.250	-8.419	1261.75	
2	1.100E+05	0.000	-0.002	229.50	
3	1.100E+05	0.250	2.655	627.20	
4	1.100E+05	0.500	3.670	573.10	
5	1.100E+05	0.738	8.419	945.10	
6	1.100E+05	0.750	8.943	987.10	
7	1.100E+05	0.770	9.500	1032.80	
8	1.100E+05	0.788	9.741	1054.80	
9	4.623E+04	1.000	12.079	1261.75	
10	2.541E+04	1.250	10.711	1261.75	
11	1.569E+04	1.500	9.040	1261.75	
12	1.044E+04	2.000	6.770	1261.75	
13	6.435E+03	2.250	5.105	1261.75	Crashes ~2400sec
14	6.235E+03	2.500	5.551	1261.75	Crashes ~2400sec
15	5.930E+03	3.000	4.341	1261.75	Crashes ~2400sec
		Max Δi =	12.079		

Crashes ~2400sec

Crashes ~2400sec

Crashes ~2400sec

		LABVIEW			
		Run 2			
Parameters					
ao (km) = 6619.25					
eo = 0.025					
theta (deg) = Max L/D		Best result from Run 1			
Δt (sec) = 5.0		fixed step			
Flight	Total Time (sec)	phi (deg)	Δi (deg)	Fuel (kg)	
16	1.100E+05	0.0	-0.002	681.80	
17	1.100E+05	10.0	1.037	689.20	
18	1.100E+05	20.0	2.238	734.10	
19	1.100E+05	30.0	3.672	796.10	
20	1.100E+05	40.0	6.449	995.70	
21	1.100E+05	45.0	8.686	1157.50	
22	9.282E+05	50.0	10.557	1261.75	
23	6.160E+05	55.0	11.385	1261.75	
24	4.623E+04	60.0	12.071	1261.75	
25	3.600E+04	65.0	12.660	1261.75	
26	2.624E+04	70.0	13.298	1261.75	
27	1.683E+04	75.0	13.426	1261.75	
28	1.188E+04	80.0	14.128	1261.75	
29	7.225E+03	85.0	14.585	1261.75	
30	2.750E+03	90.0	14.552	1261.75	Crashes ~2450sec
		Max Δi =	14.585		
	Best Roll angle	85.0			

		LABVIEW				
		Run 3				
Parameters						
ao (km) = 6619.25						
eo = 0.025						
phi (deg) = 85.0			Best result from Run 2			
Δt (sec) = 5.0			fixed step			
		Normalized				
Flight	Total Time (sec)	theta	Δi (deg)	Fuel (kg)		
31	2.780E+03	-0.34	-7.452	1261.75		
32	1.100E+05	0.00	-0.001	241.1		
33	8.020E+03	0.34	8.430	1261.75		
34	1.261E+04	0.68	13.278	1261.75		
35	1.208E+04	1.00	14.847	1261.75		
36	1.205E+04	1.02	14.824	1261.75		
37	1.201E+04	1.04	14.802	1261.75		
38	1.194E+04	1.07	14.654	1261.75		
39	7.225E+03	1.36	14.605	1261.75		
40	6.915E+03	1.69	13.414	1261.75	Crashes ~2450sec	
41	6.440E+03	2.37	10.404	1261.75	Crashes ~2450sec	
42	6.065E+03	3.05	6.667	1261.75	Crashes ~2400sec	
43	2.505E+03	4.07	3.564	1261.75	Crashes ~2300sec	
		Max Δi =	14.847			

		LABVIEW				
		Run 4				
Parameters						
ao (km) =	6600	Hp (km) =	64.00			
eo =	0.025	Ha (km) =	394.00			
theta (deg) =	Max L/D	Best result from Labview Run 3				
Δt (sec) =	5.0	fixed-step				
Flight	Total Time (sec)	phi (deg)	Δi (deg)	Fuel (kg)		
44	2.050E+04	65.00	13.292	1261.75		
45	1.575E+04	70.00	13.971	1261.75		
46	1.122E+04	75.00	14.546	1261.75		
47	6.840E+03	80.00	14.756	1261.75	Crashes ~2400sec	
48	6.600E+03	85.00	15.850	1261.75	Crashes ~2400sec	
		Max Δi =	15.850			
	Best Roll angle	85.00				

		LABVIEW				
		Run 5				
Parameters						
ao (km) =	6617	Hp (km) =	60.72			
eo =	0.028	Ha (km) =	431.28			
theta (deg) =	Max L/D	Interp diff b/w Run 1 & Run 3				
Δt (sec) =	5.0	fixed-step				
Flight	Total Time (sec)	phi (deg)	Δi (deg)	Fuel (kg)		
49	2.030E+04	65.00	13.186	1261.75		
50	1.571E+04	70.00	13.926	1261.75		
51	1.118E+04	75.00	14.462	1261.75		
52	6.805E+03	80.00	15.024	1261.75	Crashes ~2400sec	
53	6.600E+03	85.00	15.315	1261.75	Crashes ~2400sec	
		Max Δi =	15.315			
	Best Roll angle	85.00				

Appendix M - SIMULINK™ Flight Spreadsheets

		Simulink			
		Run 1			
Parameters					
ao (km) =	6619.25	Hp (km) =	82.76875		
eo =	0.025	Ha (km) =	413.73125		
phi (deg) =	60	Arbitrary starting point			
Dt (sec) =	5	fixed-step			
		Normalized			
Flight	Total Time (sec)	theta	Δi (deg)	Fuel (kg)	
9a	2.500E+05	-0.317	-2.032	666.80	
9b	2.000E+05	0.000	0.075	574.20	
9c	2.000E+05	0.317	3.138	538.70	
9d	2.000E+05	0.635	6.886	729.10	
9e	2.000E+05	0.937	12.565	1133.00	
9f	2.000E+05	0.952	12.884	1157.00	
9g	2.000E+05	0.978	13.453	1201.00	
9h	2.000E+05	1.000	14.084	1250.00	
9i	7.867E+04	1.270	13.700	1261.75	
9j	4.108E+04	1.587	11.964	1261.75	
9k	1.604E+04	2.222	8.329	1261.75	
9l	6.856E+03	2.857	4.260	1261.75	Crashed
9m	2.427E+03	3.492	1.347	1261.75	Crashed
		Max Δi =	14.084		

Crashed
Crashed

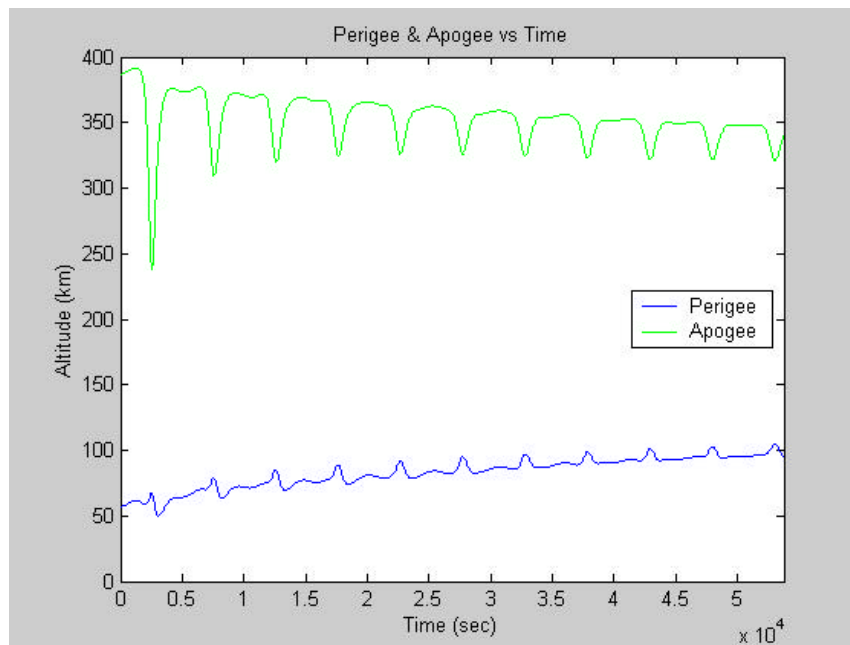
SIMULINK				
Run 2				
Parameters				
ao (km) = 6619.25				
eo = 0.025				
theta (deg) = Max L/D	Best pitch from Matlab Run 1			
Δt (sec) = 5.0	fixed-step			
Flight	Total Time (sec)	phi (deg)	Δi (deg)	Fuel (kg)
10a	1.000E+05	0.0	0.000	598.4
10b	1.100E+05	10.0	1.313	630.9
10c	1.100E+05	20.0	2.668	649.6
10d	1.100E+05	30.0	4.113	682.9
10e	1.100E+05	40.0	5.728	735.5
10f	1.100E+05	45.0	6.640	771.7
10g	1.100E+05	50.0	7.657	816.8
10h	1.100E+05	55.0	8.818	873.4
10i	1.100E+05	60.0	10.187	945.4
10j	1.100E+05	65.0	11.865	1039.0
10k	1.100E+05	70.0	14.025	1164.0
10l	9.791E+04	75.0	15.860	1261.75
10m	7.636E+04	80.0	16.205	1261.75
10n	5.578E+04	85.0	16.404	1261.75
10o	3.981E+04	90.0	16.403	1261.75
		Max Δi =	16.404	
Best Roll angle		85.0		

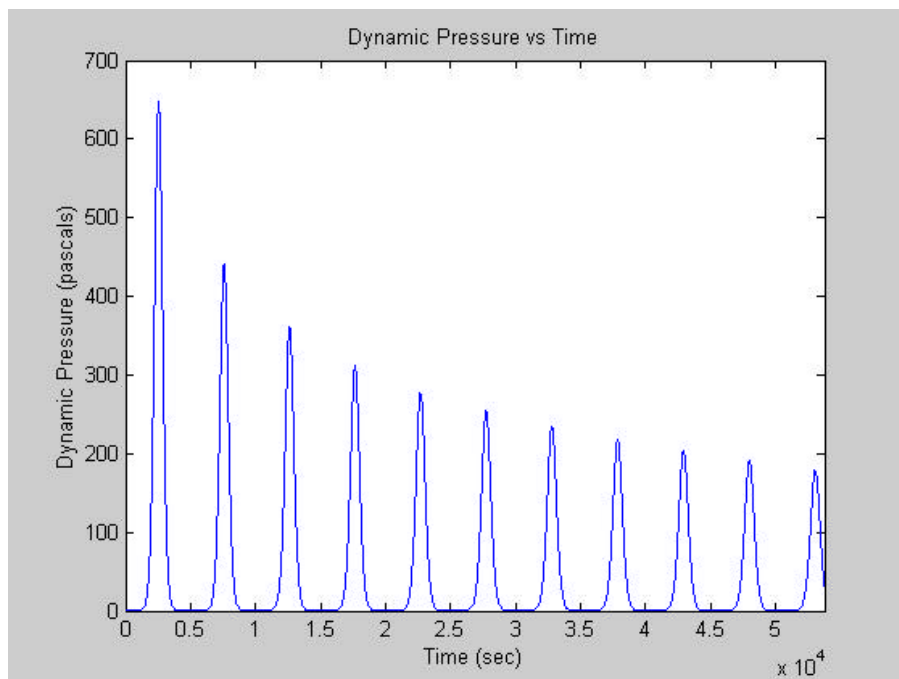
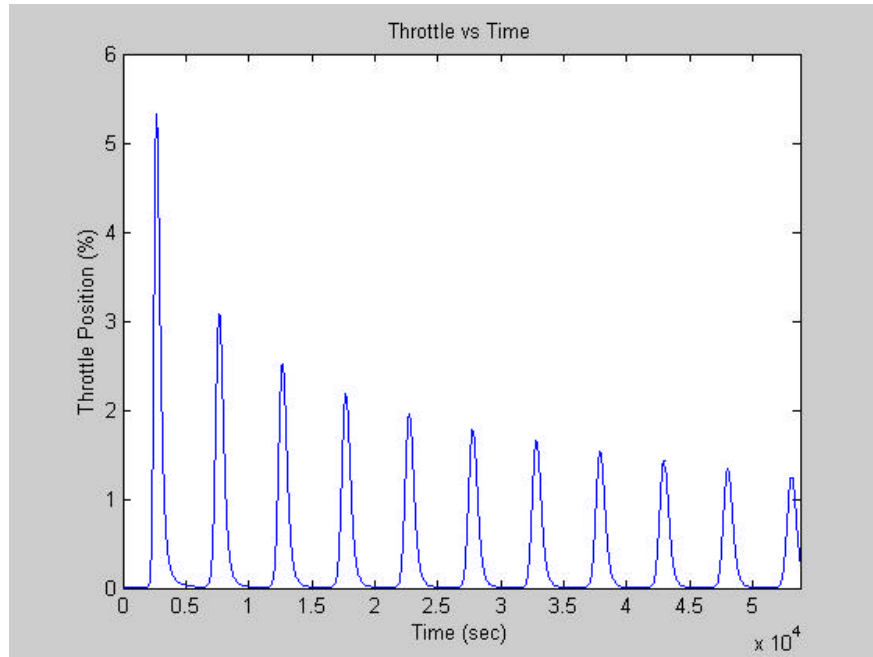
		SIMULINK			
		Run 3			
Parameters					
ao (km) = 6619.25					
eo = 0.025					
phi (deg) = 85.0		Best result from Matlab Run 2			
Δt (sec) = 5.0		fixed-step			
		Normalized			
Flight	Total Time (sec)	theta	Δi (deg)	Fuel (kg)	
11a	1.100E+05	-0.333	-1.653	485.00	
11b	1.100E+05	0.000	0.081	511.90	
11c	1.100E+05	0.333	3.745	556.80	
11d	1.100E+05	0.667	9.684	874.50	
11e	6.790E+04	0.983	16.339	1261.75	
11f	6.646E+04	1.000	16.428	1261.75	
11g	6.101E+04	1.027	16.423	1261.75	
11h	5.577E+04	1.050	16.404	1261.75	
11i	2.848E+04	1.333	15.816	1261.75	
11j	1.743E+04	1.667	13.849	1261.75	
11k	7.517E+03	2.333	9.579	1261.75	
11l	2.640E+03	3.000	3.358	661.20	Crashed
11m	2.412E+03	3.667	1.958	521.10	Crashed
		Max Δi =	16.428		

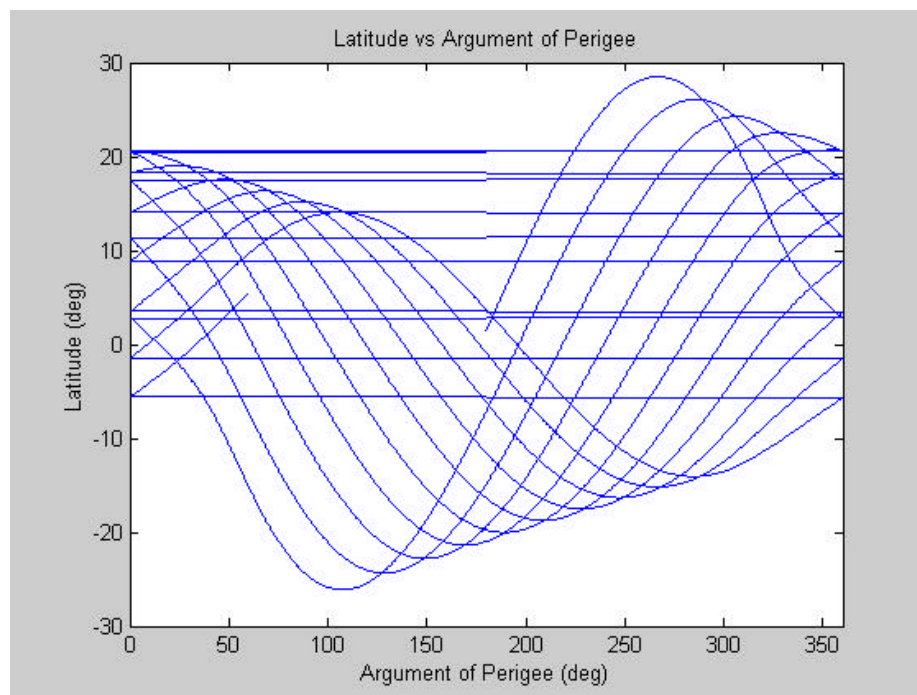
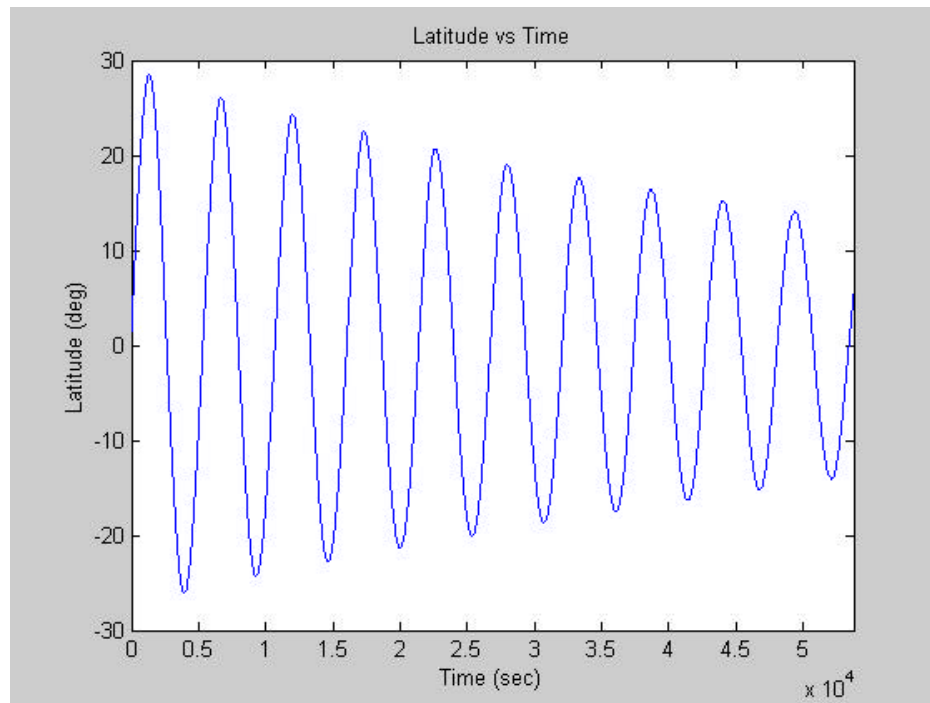
		SIMULINK		
		Run 4		
Parameters				
ao (km) = 6600		Hp (km) = 64.00		
eo = 0.025		Ha (km) = 394.00		
theta (deg) = Max L/D		Best result from Matlab Run 3		
Δt (sec) = 5.0		fixed-step		
Flight	Total Time (sec)	phi (deg)	Δi (deg)	Fuel (kg)
12a	6.834E+04	65.00	14.954	1261.75
12b	5.383E+04	70.00	15.462	1261.75
12c	4.324E+04	75.00	15.962	1261.75
12d	3.323E+04	80.00	16.294	1261.75
12e	2.338E+04	85.00	16.527	1261.75
		Max Δi =	16.527	
	Best Roll angle	85.00		

		SIMULINK		
		Run 5		
Parameters				
ao (km) =	6617	Hp (km) =	60.72	
eo =	0.028	Ha (km) =	431.28	
theta (deg) =	Max L/D	Avg from Run 1 & Run 3		
Δt (sec) =	5.0	fixed-step		
Flight	Total Time (sec)	phi (deg)	Δi (deg)	Fuel (kg)
13a	6.356E+04	65.00	14.930	1261.75
13b	4.801E+04	70.00	15.437	1261.75
13c	3.869E+04	75.00	15.900	1261.75
13d	2.874E+04	80.00	16.206	1261.75
13e	2.311E+04	85.00	16.454	1261.75
		Best Value	16.454	
	Best Roll angle	85.00		

Appendix N – Flight Simulation Plots







INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. Dr. Stephen A. Whitmore
NASA Dryden Flight Research Center
Edwards, California
4. Dr. Michael G. Spencer
Orbital Sciences Corporation
Ashburn, Virginia
5. Frank W. Cutler
NASA Dryden Flight Research Center
Edwards, California
6. Vance D. Brand
NASA Dryden Flight Research Center
Edwards, California
7. James L. Crawford
NASA Dryden Flight Research Center
Edwards, California
8. LT John P. Pienkowski
Naval Postgraduate School
Monterey, California